Gyanmanjari Innovative University

Syllabus
Gyanmanjari College Of Computer Application
Semester-1 (MCA)

Subject : Relational Database Management System - MCA2XX11301
Type of course : Skill Enhancement Course
Prerequisite : Basic knowledge of Programming

Rationale :
This course introduces the fundamentals of Database Management Systems (DBMS) with a focus on Oracle SQL and PL/SQL. It covers database concepts, SQL querying, normalization, transaction management, and data integrity. Students gain practical experience with PL/SQL, stored procedures, triggers, user management, and security. By the end, learners understand database design, implementation, and administration using Oracle/MySQL tools, combining both theoretical knowledge and practical skills.

Teaching and Examination Scheme :

Teaching Scheme			Credits	Examination Marks		Total Marks
CI	T	P	C	SEE	CCE	
2	0	2	3	100	50	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; CCE-Continuous and Comprehensive Evaluation.

Course Content:

Sr. No	Course Content	Hrs.	% Weightage
1	**Database Fundamentals and SQL Operations** **Theory Topics** : Introduction to Data, Information, and Database **SQL Command : Data Definition Language (DDL) :** CREATE, ALTER, DROP, **Data Manipulation Language (DML):** INSERT, UPDATE, DELETE, **Data Query Language (DQL) :** SELECT, DISTINCT , WHERE , ORDER BY, **Transaction Control Language (TCL) :** COMMIT, ROLLBACK ,SAVEPOINT, **Data Control Language (DCL) :** GRANT ,REVOKE, **SQL Operators:** Arithmetic Operators, Relational Operators, Logical Operators (AND, OR, NOT), IN, LIKE, BETWEEN, IS NULL, IS NOT NULL, **Constraints :** PRIMARY KEY , FOREIGN KEY , UNIQUE , NOT NULL , CHECK , **Set Operations :** Union, Intersect, Union ALL, Minus	12	20%

Practical :

1. Install and configure a relational database management system (MySQL/Oracle).
2. Create databases and tables using DDL commands (CREATE).
3. Modify existing database structures using ALTER command.
4. Remove database objects using DROP command.
5. Insert records into tables using INSERT command.
6. Modify existing records using UPDATE command.
7. Delete records from tables using DELETE command.
8. Retrieve data from tables using SELECT statement.
9. Perform data filtering using DISTINCT and WHERE clauses.
10. Sort query results using ORDER BY clause.
11. Implement Arithmetic Operators in SQL queries.
12. Implement Relational Operators in SQL queries.
13. Implement Logical Operators (AND, OR, NOT) in SQL queries.
14. Retrieve records using IN, LIKE, BETWEEN, IS NULL, and IS NOT NULL operators.
15. Create tables using PRIMARY KEY and UNIQUE constraints.
16. Create and implement FOREIGN KEY constraints to establish relationships between tables.
17. Apply NOT NULL, CHECK, and DEFAULT constraints on table attributes.
18. Perform transaction management using COMMIT command.
19. Implement ROLLBACK and SAVEPOINT commands for transaction control.
20. Manage user privileges using GRANT command.
21. Revoke user privileges using REVOKE command.
22. Perform data retrieval using multiple SQL operators and conditions.
23. Implement UNION operation on compatible tables.
24. Implement UNION ALL operation on compatible tables.
25. Perform INTERSECT operation on datasets.
26. Perform MINUS operation on datasets.

Evaluation Methods :

Sr. No.	Evaluation Methods	SEE	CCE
1	**SQL Escape Room Challenge :** Students solve step-by-step SQL tasks involving database creation, queries, constraints, transactions, and set	20	

	operations. Each correct solution unlocks the next level, logical thinking, and practical SQL skills in a timed, game-based environment		
2	**Enterprise Data Intelligence Challenge :** Students transform raw data into useful information through data cleaning, database design, and SQL-based analysis.		05
3	**Certified for Success :** Complete a technical online course, obtain the completion certificate, and upload the certificate to the GMIU Portal before the deadline.		05
Total		**20**	**10**

Examination Style :

SQL Escape Room Challenge [20 Marks] :
Faculty will provide a series of SQL-based challenges related to database creation, data manipulation, data retrieval, constraints, transactions, and set operations. Each challenge represents a level of the escape room. Students must successfully complete the current SQL task to unlock the clue or instruction for the next level. Tasks may include creating tables, inserting records, writing SELECT queries, applying constraints, using operators, managing transactions, and performing set operations. Teams must solve all levels within the allotted time and demonstrate the correctness of their SQL queries and outputs. This activity promotes collaborative learning, problem-solving skills, logical thinking, and practical application of SQL concepts in an engaging and competitive environment.

Enterprise Data Intelligence Challenge [05 Marks] :
The faculty will provide raw datasets collected from multiple sources such as Excel sheets, CSV files, online forms, or business records. The datasets may contain duplicate entries, missing values, inconsistent formats, and other data quality issues. Students will analyze the datasets to distinguish between raw data and meaningful information, perform data cleaning and organization tasks, and transform the data into useful reports. They will then design an appropriate database structure for storing the cleaned data and use SQL queries to generate insights, summaries, and analytical reports. Students will document their

	findings and submit the database design, SQL queries, and generated reports on GMIU portal. **Certified for Success [05 Marks] :** Students are required to participate in a relevant technical online course to enhance their professional knowledge, technical skills, and industry readiness. After successfully completing the course, students must download the completion certificate and upload it to the GMIU Portal within the specified deadline.		
2	**Advanced SQL Querying and Data Analysis** **Theory Topics : SQL Functions :** Numeric Functions, Aggregate Functions, String Functions, Date Functions. **Advanced Query Processing :** CASE Expressions ,NULL Handling Functions (COALESCE, NVL, NULLIF), **Window Functions:** ROW_NUMBER(), RANK(), DENSE_RANK(), LEAD(), LAG(), NTILE(),PARTITION BY and ORDER BY in Window Functions, **Types of SQL Joins:** INNER, LEFT, RIGHT, FULL OUTER, CROSS, SELF JOIN, **Subqueries**: Single Row Subqueries , Multiple Row Subqueries , Correlated Subqueries , EXISTS and NOT EXISTS,**Common Table Expressions (CTE) :**WITH Clause ,CTE based Query Design, **Database Views** : CREATE VIEW, DROP VIEW, Virtual Tables management, and Data Security. **Practical :** 27. Perform basic Numeric Functions (ABS, CEIL, FLOOR, ROUND, TRUNC) on financial database tables. 28. Implement Aggregate Functions (SUM, AVG, MIN, MAX, COUNT) to generate executive summary reports. 29. Apply advanced String Functions (CONCAT, SUBSTR, LENGTH, INSTR, LPAD, RPAD) for customer data cleaning. 30. Utilize Date Functions (ADD_MONTHS, MONTHS_BETWEEN, NEXT_DAY, LAST_DAY) to calculate project deadlines. 31. Write SQL queries using conditional CASE Expressions to dynamically categorize customer loyalty tiers. 32. Implement NULL Handling Functions (COALESCE, NVL, NULLIF) to replace missing commission fields with base values. 33. Apply Window Functions like ROW_NUMBER(), RANK(), and DENSE_RANK() to compute student test score rankings. 34. Implement Window Functions like LEAD(), LAG(), and NTILE() to analyze month-on-month sales growth	12	20%

fluctuations.

35. Perform comprehensive data analysis using PARTITION BY and ORDER BY clauses within window calculations.
36. Implement INNER JOIN and LEFT OUTER JOIN to map customer profiles with their transaction history tables.
37. Implement RIGHT OUTER JOIN, FULL OUTER JOIN, CROSS JOIN, and SELF JOIN to build organizational hierarchy maps.
38. Write nested Single Row and Multiple Row Subqueries using IN, ANY, and ALL to filter complex transactional data.
39. Implement Correlated Subqueries using EXISTS and NOT EXISTS operators for dependency validation.
40. Design and execute highly optimized queries using Common Table Expressions (CTE) using the WITH clause.
41. Create, update, and drop database Views to implement custom security layers over structural tables.

Evaluation Methods :

Sr. No.	Evaluation Methods	SEE	CCE
1	**SQL Dashboard Builder Challenge :** Students explore relationships between multiple tables and apply different SQL joins,functions to retrieve meaningful information while developing analytical and query-writing skills.	20	
2	**SQL Puzzle Race :** Students solve SQL-based puzzles by analyzing tables and generating the required outputs using advanced SQL concepts.		10
Total		**20**	**10**

Examination Style :

SQL Dashboard Builder Challenge [20 Marks] :
Faculty will provide multiple related database tables along with different business scenarios and a real-world dataset. The scenarios will require students to identify and apply appropriate Students will analyze table relationships and determine the most suitable join type to generate the required output. They will execute SQL queries, compare different join results, and explain why a particular join was selected. Students will present the generated outputs and explain the SQL logic used to create each report.

	SQL Puzzle Race [10 Marks] : Faculty will provide a set of database tables, sample records, expected outputs, and SQL-related puzzles involving Functions, Joins, Subqueries, Window Functions, and CTEs. Students will analyze the given information, identify the appropriate SQL concepts required, write the necessary queries, and generate the expected outputs. And Submit Query, Screenshot or Output in PDF Report upload on GMIU Portal.		
3	**Database Programming and Automation with PL/SQL** **Theory Topics : PL/SQL Block Structure:** DECLARE, BEGIN, EXCEPTION, END Variables, Constants, Data Types, %TYPE, %ROWTYPE, **Control Structures:** IF-THEN-ELSE, CASE, LOOP, FOR LOOP, WHILE LOOP, **Stored Procedures:** IN, OUT, IN OUT Parameters. **Stored Functions:** Return Values and Deterministic Functions. **Packages:** Package Specification and Package **Body Triggers:** DML Triggers (BEFORE/AFTER INSERT/UPDATE/DELETE), Statement-Level vs. Row-Level Triggers, DDL and System-Level Triggers, **Exception Handling:** Predefined and User-Defined Exceptions,RAISE, RAISE_APPLICATION_ERROR,MySQL Stored Programs: Delimiter, CALL syntax differences. **Practical :** 42. Create and execute a basic PL/SQL block using DECLARE, BEGIN, EXCEPTION, and END sections. 43. Write PL/SQL programs using Variables, Constants, and different Data Types. 44. Implement the use of %TYPE and %ROWTYPE attributes in PL/SQL programs. 45. Develop PL/SQL programs using IF-THEN, IF-THEN-ELSE, and Nested IF statements. 46. Create PL/SQL programs using CASE control structures. 47. Implement LOOP statements for repetitive processing. 48. Develop programs using FOR LOOP and WHILE LOOP constructs. 49. Create Stored Procedures with IN parameters. 50. Create Stored Procedures with OUT parameters. 51. Develop Stored Procedures using IN OUT parameters. 52. Create and execute Stored Functions with return values. 53. Implement Deterministic Functions and analyze their usage. 54. Create a Package Specification and Package Body for related procedures and functions.	12	20%

55. Develop and execute DML Triggers for INSERT operations.
56. Develop and execute DML Triggers for UPDATE operations.
57. Develop and execute DML Triggers for DELETE operations.
58. Compare and implement Statement-Level and Row-Level Triggers.
59. Create and test DDL Triggers for database object modifications.
60. Implement System-Level Triggers for database events.
61. Handle predefined exceptions using PL/SQL Exception Handling mechanisms.
62. Create and handle User-Defined Exceptions using RAISE statements.
63. Implement error handling using RAISE_APPLICATION_ERROR.
64. Develop a PL/SQL application integrating Procedures, Functions, Packages, and Triggers.
65. Create MySQL Stored Procedures using DELIMITER syntax.
66. Execute Stored Procedures using the CALL statement and study syntax differences between Oracle PL/SQL and MySQL Stored Programs.

Evaluation Methods :

Sr. No.	Evaluation Methods	SEE	CCE
1	**Banking Transaction Simulator :** Students develop and demonstrate a banking transaction management solution using Procedures, Functions, Triggers, and Exception Handling.	20	
2	**Logic Puzzle Lab :** Students solve PL/SQL logic and programming puzzles by applying procedural programming and problem-solving skills.		10
	Total	**20**	**10**

Examination Style:

Banking Transaction Simulator [20 Marks] :
Faculty will provide a banking database case study containing customer accounts, transaction records, and business requirements such as deposits, withdrawals, fund transfers, and transaction

	auditing. Students will analyze the given scenario and develop PL/SQL solutions using PL/SQL Blocks, Variables, Control Structures, Stored Procedures, Functions, Triggers, and Exception Handling mechanisms to automate banking operations, maintain transaction logs, handle errors, execute the programs, and demonstrate the generated outputs with proper justification of the implemented logic. **Logic Puzzle Lab [10 Marks] :** Faculty will provide various PL/SQL programming challenges, incomplete code snippets, debugging tasks, and logical problem statements involving Variables, Loops, Procedures, Functions, Triggers, and Exception Handling. Students will analyze the logic, complete or correct the code, predict outputs, and develop appropriate PL/SQL solutions while explaining their reasoning and implementation approach. After successfully completing the activity, students must upload their completed work, report, or certificate (if applicable) to the GMIU Portal.		
4	**Database Design, Modeling and Normalization** **Theory Topics : DBMS Architecture:** 3-Tier and 3-Schema Architecture, Oracle Apex**Functional Dependencies:** Full, Partial, Transitive, Normalization: 1NF, 2NF, 3NF, BCNF, **Higher Normal Forms**: 4NF, 5NF (Project-Join NF), **RelationalData Model**: Relations, Tuples, Attributes, Domains, **ER Modeling:** Entities, Attributes, Relationships, Cardinalities, Strong Entity, Weak Entity Generalization, Specialization, Aggregation, Data Anomalies , Database Design Process. **Practical :** 67. Study and compare 3-Tier Architecture and 3-Schema Architecture of DBMS. 68. Explore the Oracle APEX environment and create a simple database application. 69. Identify Functional Dependencies including Full, Partial, and Transitive Dependencies from given relations. 70. Convert unnormalized data into First Normal Form (1NF). 71. Perform normalization up to Second Normal Form (2NF). 72. Perform normalization up to Third Normal Form (3NF). 73. Convert database schemas into Boyce-Codd Normal Form (BCNF). 74. Analyze database schemas for Fourth Normal Form (4NF) and Fifth Normal Form (5NF). 75. Design Entity-Relationship (ER) Diagrams using Entities, Attributes, and Relationships. 76. Implement Cardinality Constraints and Relationship Mapping	12	20%

in ER Models.

77. Design Strong Entities and Weak Entities using ER Modeling tools.
78. Demonstrate Generalization, Specialization, and Aggregation concepts through ER diagrams.
79. Identify and resolve Insertion, Update, and Deletion Anomalies in database tables.
80. Perform complete Database Design Process for a real-world case study.

Evaluation Methods :

Sr. No.	Evaluation Methods	SEE	CCE
1	**Anomaly Detective :** Students identify database anomalies and redesign the database structure using normalization techniques to improve data integrity.	20	
2	**Database Doctor :** Students identify and fix database design problems by improving structure, relationships, and normalization.		10
Total		**20**	**10**

Examination Style:

Anomaly Detective [10 Marks] :
Faculty will provide poorly designed database tables along with sample records containing insertion, update, and deletion anomalies. Students will analyze the given tables, identify various anomalies and their causes, explain the impact on data consistency, perform appropriate normalization techniques to improve the database design, and present the redesigned schema with proper justification for eliminating the identified anomalies.

Database Doctor [10 Marks]
Faculty will provide poorly designed databases containing redundancy, anomalies, missing relationships, normalization issues, and inefficient table structures. Students will act as Database Doctors by diagnosing the problems, identifying design flaws, applying normalization techniques, improving relationships and constraints, and proposing a healthier database design with proper justification for the changes made. After completing the activity, students must upload their final solution/report to the GMIU Portal.

5	**NoSQL Database Management and MongoDB Development** **Theory Topics : Introduction to NoSQL Databases** : Introduction to NoSQL Databases. Limitations of Relational Databases. Need for NoSQL Databases. **MongoDB Fundamentals** :Introduction to MongoDB. Features of MongoDB. MongoDB Architecture. BSON Data Format. Installing MongoDB. MongoDB Compass Introduction. MongoDB Shell Basics. **Data Modeling in MongoDB** : Collections and Documents. Embedded Documents. Referencing. **CRUD Operations in MongoDB** :Insert Operations. Find Operations. Update Operations. Delete Operations. Query Operators. **Aggregation and Indexing** : Aggregation Framework. Aggregation Pipeline. Indexing Concepts. **Practical:** 81. Install MongoDB and configure the MongoDB environment using MongoDB Compass and MongoDB Shell. 82. Create databases and collections and perform basic database operations using MongoDB Shell. 83. Design MongoDB collections and document structures using Embedded Documents and Referencing techniques. 84. Perform CRUD Operations (Insert, Find, Update, and Delete) on MongoDB collections. 85. Apply Query Operators for data filtering and retrieval. 86. Perform data analysis using the Aggregation Framework and Aggregation Pipelines. 87. Create Indexes and evaluate their impact on query performance. 88. Design and implement a MongoDB database for a real-world application. 89. Integrate CRUD Operations, Query Operators, Aggregation Pipelines, and Indexing concepts in the designed application. 90. Demonstrate the complete MongoDB application with appropriate reports and outputs. **Evaluation Methods :** **Sr. No.** | **Evaluation Methods** | **SEE** | **CCE** 1 | **MongoDB Mini Project** : Students design and implement a complete MongoDB-based application using data modeling, CRUD operations, | 20 |	12	20%

	aggregation, and indexing concepts.		
2	**Startup NoSQL Architect :** Students design and present a MongoDB-based database architecture for a startup by applying NoSQL data modeling concepts.		10
Total		**20**	**10**

Examination Style :

MongoDB Mini Project [20 Marks] :
The faculty will provide a real-world case study such as an E-Commerce System, Online Learning Platform, Hospital Management System, or Food Delivery Application with specific data management requirements. Students will design MongoDB collections and document structures, perform CRUD operations, implement query and update operators, create aggregation pipelines, apply indexing techniques for performance optimization, and demonstrate a complete working MongoDB solution while explaining their database design and implementation decisions.

Startup NoSQL Architect [10 Marks] :
Faculty will provide startup ideas such as Food Delivery Systems, Ride Booking Applications, Online Learning Platforms or E-Commerce Systems along with their business requirements. Students will act as NoSQL Architects by designing MongoDB collections, document structures, embedded documents, references, indexing strategies, and aggregation requirements suitable for the given startup. Students will present and justify their database design decisions. After completing the activity, students must upload their final work/report to the GMIU Portal

Suggested Specification :

Distribution of Theory Marks

(Revised Bloom's Taxonomy)

Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	10%	10%	20%	10%	10%	40%

Note : This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcome :

After learning the course the students should be able to:	
CO1	Demonstrate a comprehensive understanding of fundamental database concepts and Oracle SQL syntax, enabling the design, querying, and effective management of relational database systems.
CO2	Exhibit proficiency in advanced SQL querying techniques, structural analytical window functions,dynamic reporting logic, and virtual safehouse database views to ensure efficient and consistent data management.
CO3	Utilize PL/SQL programming constructs, including control structures, explicit implicit cursors with attributes, and comprehensive exception handling mechanisms to develop robust, secure, and efficient database-driven applications.
CO4	Examine the architecture of enterprise databases (3-Tier/3-Schema), ER modeling components, and apply data integrity normalization mechanisms from INF up to BCNF to resolve design anomalies.
CO5	Implement advanced NoSQL functionalities and MongoDB fundamentals, including embedded/referenced data modeling, CRUD execution layers,aggregation pipelines, and indexing optimization strategies for modern application architectures.

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students.
A minimum of 20–30% of the course content shall be suggested to deliver through Flipped Learning, where students study digital learning resources before the classroom session.Classroom time shall be utilized for discussion, problem-solving, practical applications, and higher-order learning activities.

Students shall be encouraged to use AI-powered learning platforms, simulation software,virtual laboratories, digital libraries, research databases, online certification courses, NPTEL,SWAYAM, MOOCs, and other technology-enabled learning resources to strengthen conceptual understanding and practical competence.

Continuous assessment shall be conducted through Active Learning Assignments (ALAs),quizzes, skill demonstrations, presentations, practical exercises, projects, case studies,innovation activities, peer assessment, reflective learning, and industry-oriented tasks instead of relying primarily on conventional written assignments.

Reference Books:

[1] Database Systems: A Practical Approach to Design, Implementation, and Management, Thomas Connolly, Carolyn Begg, Pearson Publication.
[2] RDBMS In-Depth: Mastering SQL and PL/SQL Concepts, Dr. Madhavi Vaidya, BPB Publications
[3] Oracle Database 12c PL/SQL Programming, Michael McLaughlin, McGraw Hill
[4] Database System Concepts, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw Hill
[5] MongoDB 8.0 in Action (3rd Edition), Hans-Jürgen Schönig, Manning Publications

Suggested Assessment Guidelines:

Module 1: Database Fundamentals and SQL Operations

Program Implementation Task(20Marks):		
Criteria	**Description**	**Marks**
Database Design	Create database, tables, and apply appropriate constraints.	5
SQL Implementation	Execute DDL, DML, and DQL commands correctly.	5
Problem Solving	Apply transactions, set operations, and complete challenge levels logically.	5
Output & Documentation	Generate correct outputs and provide proper documentation/explanation.	5
Total		**20**

Module 2: Advanced SQL Querying and Data Analysis

Program Implementation Task(20Marks):		
Description	**Description**	**Marks**
Database Relationships	Design related tables with appropriate keys and relationships.	5
SQL Query Development	Use joins, aggregate functions, and built-in SQL functions correctly.	5
Data Analysis & Reporting	Generate meaningful reports using filtering, grouping, and sorting.	5
Output & Presentation	Produce accurate results with proper query formatting and documentation.	5
Total		**20**

Module 3: Database Programming and Automation with PL/SQL

Program Implementation Task(20Marks):		
Description	**Description**	**Marks**
Stored Procedures & Functions	Develop procedures and user-defined functions for banking operations.	5
Triggers & Exception Handling	Implement triggers and handle runtime exceptions effectively.	5
Transaction Management	Perform deposits, withdrawals, and fund transfers while maintaining data consistency.	5
Testing & Documentation	Demonstrate successful execution and document the implementation.	5
Total		**20**

Module 4: Database Design, Modeling and Normalization

Program Implementation Task(20Marks):		
Description	**Description**	**Marks**
Anomaly Identification	Identify insertion, update, and deletion anomalies in the database.	5
Normalization	Apply 1NF, 2NF, and 3NF correctly to redesign the database.	5
Database Redesign	Create an improved schema ensuring integrity and reduced redundancy.	5
Documentation & Justification	Clearly explain the normalization process and final design.	5
Total		**20**

Module 5 : NoSQL Database Management and MongoDB Development

Program Implementation Task(20Marks):		
Description	**Description**	**Marks**
Data Modeling	Design collections and define appropriate document structures.	5
CRUD Operations	Implement Create, Read, Update, and Delete operations successfully.	5
Aggregation & Indexing	Apply aggregation pipelines and indexing for efficient data retrieval.	5
Project Demonstration	Demonstrate application functionality with proper documentation and presentation.	5
Total		**20**