Gyanmanjari Innovative University

Syllabus
Gyanmanjari Diploma Engineering College
Semester-3 (Diploma)

Subject: Efficient Data Handling with DBMS – DET2CE13204

Type of course: Professional Core

Prerequisite: Students should have basic knowledge of computer operations, file handling, and simple data storage using tables or spreadsheets. They should also have basic logical thinking ability to identify real-world data, records, fields, and relationships.

Rationale:
In today's digital environment, every software application depends on efficient data storage, retrieval, security, and management. This subject, Efficient Data Handling with DBMS, is designed to help Diploma Semester 3 students understand database concepts through practical problem-solving rather than only theoretical study. Students will learn how real-world data problems such as duplication, inconsistency, slow searching, invalid entries, poor design, unsafe transactions, and unauthorized access can be solved using DBMS concepts, ER modeling, SQL, constraints, normalization, transactions, backup, and security mechanisms. This practical approach will prepare students to design and implement reliable database solutions for academic, business, and industry-oriented applications.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks		Total Marks
CI	T	P	C	SEE	CCE	
4	0	2	5	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	**Fundamentals of Database Management Systems** ***Theory Topics:*** Introduction to data, information and database, need of Database Management System, file system vs DBMS, advantages and limitations of DBMS, DBMS users and roles, DBMS architecture with 1-tier, 2-tier and 3-tier architecture, concept of data independence including logical and physical data independence, database schema and instance, introduction to data models including hierarchical model, network model, relational model and object-oriented model, and basic relational database concepts such as relation, tuple, attribute, domain, degree and cardinality. ***Practical:*** 1. Data Identification: Identify data, information, and database from given real-world student records. 2. File-Based Data Handling: Analyse problems of storing student records in Excel/CSV/text files. 3. DBMS Need Analysis: Compare manual file-based storage with DBMS-based storage for searching, updating, and deleting records. 4. Basic Database Creation: Create a basic college database for storing student information. 5. Table Creation: Create a Student table with suitable fields and data types. 6. Data Insertion: Insert multiple student records into the Student table. 7. Basic Data Retrieval: Retrieve all student records from the database using basic SQL queries. 8. Schema and Instance Identification: Identify database schema and instance using a created Student table. 9. Relational Concepts Identification: Identify relation, tuple, attribute, domain, degree, and cardinality from a database table. 10. DBMS User Role Analysis: Identify different DBMS users and their access requirements in a college database system. 11. DBMS Architecture Analysis: Analyse 1-tier, 2-tier, and 3-tier DBMS architecture using practical application scenarios.	18	20%

12. Data Model Comparison: Represent the same real-world data using hierarchical, network, relational, and object-oriented data models at a basic level.
13. Data Model Comparison: Represent the same real-world data using hierarchical, network, relational, and object-oriented data models at a basic level.
14. Student Database Implementation: Design and implement a basic student information database.

Examination Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	**DBMS Application Flow and Architecture Analysis** Analyse a real-world application, identify data flow and users, and select a suitable DBMS architecture.	10	-
2	**Database Structure Evaluation and Implementation** Identify database design errors, correct the structure, and implement it using basic SQL operations.	10	-
3	**ALA – Real-World Database System Analysis and Improvement Proposal** Study a real-world system, identify database-related issues, and propose suitable improvements through a presentation or visual format.	-	10
	Total	20	10

Examination Style:

1. **DBMS Application Flow and Architecture Analysis (10 Marks)**:
   - In this task, students will be given an unseen application scenario such as a student portal, library system, hospital appointment system, admission system, hostel management system, or result management system. Students are required to identify the major data requirements and different DBMS users, define their access requirements, and prepare a basic flow showing

	how data will be entered, stored, retrieved, updated, and accessed. Students will also select a suitable DBMS architecture with proper justification and explain how the proposed database system provides better data organization, accessibility, and control than a file-based approach. 2. **Database Structure Evaluation and Implementation (10 Marks)**: • In this task, students will be given an incomplete or incorrectly designed database structure containing unsuitable fields, incorrect data types, duplicated information, missing identifiers, or inconsistent records. Students are required to identify and correct the errors, redesign the database using suitable tables, fields, data types, and identifiers, and implement the corrected structure using SQL commands. Students will insert sample records, perform basic retrieval, updating, and deletion operations, and identify relation, tuple, attribute, domain, degree, cardinality, schema, and instance from the corrected database. 3. **ALA – Real-World Database System Analysis and Improvement Proposal (10 Marks):** • In this activity, students will select and study a real-world digital system such as an online shopping platform, hospital management system, banking application, university portal, railway reservation system, food-delivery application, or government service portal. Students will identify the major categories of data, different users and their access requirements, probable DBMS architecture, suitable data model, and the flow of data from entry to storage and retrieval. Students will also identify possible data-management problems and propose suitable improvements for better data organization, accessibility, consistency, or security. The activity will be represented through a presentation, concept map, infographic, poster, or demonstration and submitted through the GMIU Web Portal.		

2	**Database Design Using ER Model, Keys and Integrity Constraints** ***Theory Topics:*** Importance of database design, introduction to Entity Relationship Model, entity, entity set and attributes, types of attributes including simple, composite, single-valued, multi-valued and derived attributes, relationship and relationship set, degree of relationship including unary, binary and ternary relationship, mapping cardinality including one-to-one, one-to-many, many-to-one and many-to-many relationship, participation constraints including total and partial participation, ER diagram symbols and notations, conversion of ER diagram into relational schema, keys in DBMS including super key, candidate key, primary key, alternate key, foreign key and composite key, and integrity constraints including domain constraint, key constraint, entity integrity constraint and referential integrity constraint. ***Practical:*** 15. Entity-Based Table Design: Create separate database tables for given entities such as Student, Subject, Faculty, Department, and Book. 16. Attribute-Based Table Creation: Create tables by selecting suitable attributes and data types for each entity. 17. Composite Attribute Implementation: Create table structures by dividing composite attributes such as name and address into suitable sub-fields. 18. Multi-Valued Attribute Handling: Create separate tables to handle multi-valued attributes such as multiple mobile numbers or multiple email IDs. 19. Relationship Table Creation: Create relational tables to represent relationships between two or more entities. 20. One-to-One Relationship Implementation: Implement one-to-one relationship between two tables using primary key and foreign key. 21. One-to-Many Relationship Implementation: Implement one-to-many relationship between two tables using foreign key. 22. Many-to-Many Relationship Implementation: Implement many-to-many relationship using a junction table. 23. ER Diagram Drawing: Draw ER diagrams for given database requirements using proper ER diagram symbols and notations.	18	20%

24. ER Diagram to Table Conversion: Convert ER diagrams into relational tables.
25. Primary Key Implementation: Create tables by applying primary key constraints.
26. Foreign Key Implementation: Create related tables by applying foreign key constraints.
27. Composite Key Implementation: Create tables by applying composite keys where required.
28. Integrity Constraint Implementation: Implement domain constraint, key constraint, entity integrity constraint, and referential integrity constraint in database tables.

Examination Style:

Sr. No.	Evaluation Methods	SEE	CCE
1	**Requirement – to - ER Design Challenge** Analyse database requirements and create a complete ER diagram with suitable entities, attributes, relationships, and constraints.	10	-
2	**Relational Schema Correction and Constraint Testing** Identify schema errors, correct the relational design, and test keys and integrity constraints using sample records.	10	-
3	**ALA – Database Integrity Risk Analysis and Design Audit** Review an ER design and relational schema, trace business rules, identify design risks, and verify database constraints.	-	10
	Total	20	10

Examination Style:

1. **Requirement-to-ER Design Challenge (10 Marks)**:
   - In this task, students will be given an unseen, incomplete, or semi-structured database requirement based on a system such as course registration, library issue-return, inventory management, hospital appointment, hostel allocation, or event registration. Students are required to analyse the requirement, identify missing data elements and business rules,

	define suitable entities and attributes, and classify the attributes as simple, composite, single-valued, or multi-valued. Students will prepare a complete ER diagram showing appropriate relationships, cardinality, degree of relationship, and participation constraints. Students must also justify the major design decisions made while converting the unclear requirement into a complete conceptual database model. 2. **Relational Schema Correction and Constraint Testing (10 Marks)**: • In this task, students will be given an unseen ER diagram or an incorrectly designed relational schema containing issues such as missing keys, incorrect relationships, duplicated attributes, unsuitable data types, invalid references, or incomplete constraints. Students are required to identify and correct the design errors, convert the corrected design into relational tables, and define suitable primary keys, foreign keys, composite keys, and junction tables. Students must implement the corrected structure and test the constraints using valid and invalid sample records to demonstrate the prevention of duplicate key values, NULL primary keys, invalid foreign key references, incorrect domain values, and violations of referential integrity. 3. **ALA – Database Integrity Risk Analysis and Design Audit (10 Marks):** • In this activity, students will be provided with a complete application requirement, ER diagram, and corresponding relational schema. Students are required to prepare a business-rule traceability table showing how each requirement is represented through entities, attributes, relationships, cardinality, participation constraints, keys, and integrity constraints. Students will review whether all business rules are properly represented, identify any ambiguous or unsupported requirements, and verify selected primary key, foreign key, composite key, domain, entity integrity, and referential integrity rules using valid and invalid test records in the database software. Students will record the expected result, actual result, and verification status and explain their observations through a laboratory		

	demonstration and oral questioning under faculty supervision.		
3	**SQL Fundamentals, Data Manipulation and Report Generation** ***Theory Topics:*** Introduction to SQL, SQL data types, categories of SQL commands including DDL, DML, DQL, DCL and TCL, DDL commands such as CREATE, ALTER, DROP and TRUNCATE, DML commands such as INSERT, UPDATE and DELETE, DQL command SELECT, use of WHERE clause with comparison and logical operators, DISTINCT keyword, ORDER BY clause, BETWEEN, IN, LIKE and IS NULL operators, aggregate functions including COUNT, SUM, AVG, MIN and MAX, GROUP BY and HAVING clause, constraint-based data validation using NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK and DEFAULT, and practical use of SQL for data cleaning, filtering, correction, validation and report generation. ***Practical:*** 29. Constraint-Based Database Creation: Create database tables by selecting suitable data types and applying required constraints. 30. DDL Command Execution: Execute CREATE, ALTER, DROP, and TRUNCATE commands based on given structural requirements. 31. Table Modification Modify an existing table by adding, changing, and removing fields using ALTER command. 32. Bulk Record Insertion: Insert multiple valid records into database tables using INSERT command. 33. Invalid Record Testing: Test invalid data insertion against NOT NULL, UNIQUE, CHECK, DEFAULT, PRIMARY KEY, and FOREIGN KEY constraints. 34. Conditional Record Update: Update selected records using UPDATE command with single and multiple conditions. 35. Controlled Record Deletion: Delete specific records using DELETE command with proper WHERE conditions. 36. Selective Data Retrieval: Retrieve selected columns and filtered records using SELECT and WHERE clauses. 37. Multi-Condition Query Execution: Retrieve records using comparison operators, logical operators, BETWEEN, IN, LIKE, and IS NULL.	18	20%

38. Sorting and Duplicate Control: Retrieve sorted and unique records using ORDER BY and DISTINCT keyword.
39. Data Cleaning Using SQL: Correct incomplete, duplicate, invalid, or wrongly entered records using SQL commands.
40. Aggregate Calculation: Apply COUNT, SUM, AVG, MIN, and MAX functions on numeric database fields.
41. Grouped Report Generation: Generate category-wise reports using GROUP BY clause.
42. Conditional Group Report Generation: Generate filtered summary reports using GROUP BY with HAVING clause.
43. SQL Output Verification: Verify query outputs by comparing expected results with actual SQL results.

Examination Style:

Sr. No.	Evaluation Methods	SEE	CCE
1	**SQL Error Diagnosis and Query Reconstruction** Identify SQL errors, correct the commands, execute them, and verify the expected and actual outputs.	10	-
2	**Data Audit, Cleaning and Analytical Report Generation** Identify and clean invalid data, then generate and verify analytical reports using SQL queries and aggregate functions.	10	-
3	**ALA – SQL Validation Test Case and Data Quality Audit** Prepare SQL validation test cases, identify data-quality defects, and recommend suitable corrective actions.	-	10
	Total	20	10

Examination Style:

1. **SQL Error Diagnosis and Query Reconstruction (10 Marks):**
   - In this task, students will be given an unseen database requirement along with a set of incorrect, incomplete, or logically mismatched SQL commands related to table creation, structural modification, constraint application, record insertion, updating, deletion, filtering, and sorting. Students are required to identify syntax, logical,

	constraint-related, and execution-sequence errors, reconstruct the SQL commands, and execute them successfully. Students must also compare the expected and actual outputs and prepare a brief error-correction summary explaining the identified problems and the corrections applied. **2. Data Audit, Cleaning and Analytical Report Generation (10 Marks):** • In this task, students will be given a database containing incomplete, duplicate, invalid, inconsistent, or incorrectly entered records along with specific business-reporting requirements. Students are required to audit the available data, classify the identified data-quality issues, and correct or remove inappropriate records using suitable SQL commands. After cleaning the data, students must generate filtered, sorted, unique, aggregate, category-wise, and condition-based summary reports using SELECT, WHERE, ORDER BY, DISTINCT, aggregate functions, GROUP BY, and HAVING clauses. Students must verify the generated reports by comparing expected results with actual SQL outputs and justify the major corrections made during the data-cleaning process. **3. ALA – SQL Validation Test Case and Data Quality Audit (10 Marks):** • In this activity, students will select a real-world database application such as a student result system, employee management system, library system, hospital registration system, inventory system, or online order system. Students are required to identify the important data-entry rules and prepare a structured SQL validation test plan containing valid cases, invalid cases, expected results, actual results, and test status. The activity must include testing of selected rules related to missing values, duplicate values, incorrect ranges, invalid references, inappropriate data types, and inconsistent records. Students will prepare a defect log describing the identified data-quality issues, their possible impact, and suitable corrective recommendations. The final submission should include the selected system description, business-rule mapping, test-case table, execution evidence, defect log, output verification, and brief documentation submitted through the GMIU Web Portal.		

4	**Advanced SQL, Query Optimization and Database Normalization** ***Theory Topics:*** Need of multiple tables in database design, advanced data retrieval using joins including INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN and SELF JOIN, subqueries including single row subquery, multiple row subquery and nested subquery, set operations including UNION, UNION ALL, INTERSECT and EXCEPT/MINUS, views including creating views, updating views and advantages of views, indexing including need of index, advantages and limitations of index, introduction to query optimization, functional dependency, need of normalization, normal forms including 1NF, 2NF, 3NF and BCNF, and problems due to poor database design including insertion anomaly, update anomaly and deletion anomaly. ***Practical:*** 44. Multiple Table Creation: Create multiple related tables to store connected data using primary key and foreign key relationships. 45. Inner Join Execution: Retrieve matching records from two or more related tables using INNER JOIN. 46. Left and Right Join Execution: Retrieve matching and non-matching records from related tables using LEFT JOIN and RIGHT JOIN. 47. Full Join Implementation: Implement FULL JOIN logic using supported SQL syntax or alternative query methods. 48. Self Join Execution: Apply SELF JOIN on a table to retrieve hierarchical or self-referencing data. 49. Single Row Subquery Execution: Write subqueries that return a single value and use it in the main query. 50. Multiple Row Subquery Execution: Write subqueries that return multiple values using IN, ANY, or ALL operators. 51. Nested Subquery Execution: Execute nested subqueries to retrieve data from multiple levels of conditions. 52. Set Operation Implementation: Apply UNION, UNION ALL, INTERSECT, and EXCEPT/MINUS operations on compatible result sets. 53. View Creation: Create views to display selected and restricted data from one or more tables. 54. View Update Testing: Perform update operations on simple views and analyse limitations of updating complex views.	18	20%

55. Index Creation: Create indexes on frequently searched columns to improve data retrieval performance.
56. Query Optimization Practice: Rewrite inefficient SQL queries using selected columns, proper conditions, joins, and indexing.
57. Normalization Execution: Convert unnormalized tables into 1NF, 2NF, 3NF, and BCNF.
58. Anomaly Removal: Redesign poorly structured tables to remove insertion, update, and deletion anomalies.

Examination Style:

Sr. No.	Evaluation Methods	SEE	CCE
1	**Multi-Table Query Diagnosis and Optimization** Identify errors in multi-table SQL queries, reconstruct them using suitable techniques, and improve query efficiency.	10	-
2	**Normalization and Database Redesign Challenge** - Analyse a poorly designed table, normalize it up to 3NF or BCNF, and remove data redundancy and anomalies.	10	-
3	**ALA – Database Performance and Access-Control Audit** Analyse a database system, identify performance and access-control issues, and propose suitable views, indexes, query improvements, and restrictions.	-	10
	Total	20	10

Examination Style:

1. **Multi-Table Query Diagnosis and Optimization (10 Marks):**
   - In this task, students will be given an unseen database containing multiple related tables along with incomplete, incorrect, logically mismatched, or inefficient SQL queries. Students are required to analyse the reporting requirement, identify errors in table relationships and join conditions, select suitable joins, subqueries, nested queries, or set operations, and reconstruct the queries to generate accurate results. Students must also improve query readability and

	efficiency by using selected columns, appropriate conditions, suitable query structure, and available indexes. The generated output must be verified against the expected reporting requirement. 2. **Normalization and Database Redesign Challenge (10 Marks):** • In this task, students will be given an unnormalized or poorly structured table containing repeated data, functional dependencies, and insertion, update, or deletion anomalies. Students are required to identify candidate keys, partial dependencies, transitive dependencies, and normalization-related problems. They must decompose the given structure into suitable relations up to 3NF or BCNF, define appropriate primary and foreign keys, and prepare the redesigned relational schema. Students must justify each decomposition step and explain how the redesigned structure reduces redundancy and prevents insertion, update, and deletion anomalies. 3. **ALA – Database Performance and Access-Control Audit (10 Marks):** • In this activity, students will select an existing or proposed database-based system such as a university portal, hospital management system, inventory system, employee management system, banking application, library system, or online shopping platform. Students are required to study the reporting, data-access, and search requirements of the selected system and prepare a database performance and access-control audit. Students must propose suitable views, indexing strategies, query improvements, and access restrictions and explain their expected effect on data security, query performance, and reporting efficiency. The final submission should include the selected system description, user-access matrix, query-performance observations, identified issues, proposed improvements, supporting evidence, and brief documentation submitted through the GMIU Web Portal.		

5	**Database Transactions, Concurrency Control, Security and Recovery** ***Theory Topics:*** Transaction in DBMS, properties of transaction including Atomicity, Consistency, Isolation and Durability, transaction states, transaction control using COMMIT, ROLLBACK and SAVEPOINT, concurrency control, problems in concurrent transactions including lost update problem, dirty read problem and unrepeatable read problem, introduction to locks including shared lock and exclusive lock, database backup and recovery, database security, user management and access control, DCL commands including GRANT and REVOKE, TCL commands including COMMIT, ROLLBACK and SAVEPOINT, introduction to stored procedures, introduction to triggers, and real-world database applications including student management system, library management system, inventory management system, banking system and hospital management system. ***Practical:*** 59. Transaction Table Creation: Create database tables required for transaction-based operations. 60. Transaction Execution: Execute a complete transaction using multiple SQL operations. 61. Commit Operation Implementation: Apply COMMIT command to permanently save successful transaction changes. 62. Rollback Operation Implementation: Apply ROLLBACK command to undo failed or incorrect transaction changes. 63. Savepoint Implementation: Apply SAVEPOINT to partially rollback selected operations within a transaction. 64. ACID Property Demonstration: Demonstrate Atomicity, Consistency, Isolation and Durability through transaction-based SQL execution. 65. Concurrent Transaction Simulation: Simulate concurrent database operations on the same record. 66. Lost Update Problem Demonstration: Demonstrate lost update problem using simultaneous update operations. 67. Dirty Read Problem Demonstration: Demonstrate dirty read problem using uncommitted transaction data.	18	20%

68. Unrepeatable Read Problem Demonstration: Demonstrate unrepeatable read problem using repeated read operations.
69. Lock Implementation: Apply shared lock and exclusive lock concepts for controlled data access.
70. Database Backup Creation: Create backup of an existing database using database backup utilities or commands.
71. Database Recovery Execution: Restore database from backup after accidental deletion or data loss.
72. User Creation and Access Control: Create database users and assign role-based access permissions.
73. Grant and Revoke Implementation: Apply GRANT and REVOKE commands to manage database privileges.
74. Stored Procedure Creation: Create stored procedures for repeated database operations.
75. Trigger Creation: Create triggers for automatic log entry or rule-based database action.

Examination Style:

Sr. No.	Evaluation Methods	SEE	CCE
1	**Transaction Failure Diagnosis and Recovery Challenge** Identify transaction failures, apply suitable recovery commands, and analyse concurrency problems and ACID properties.	10	-
2	**Security, Procedure and Trigger Automation** Correct user privileges, stored procedures, and triggers, and verify database security and automated operations.	10	-
3	**ALA – Database Continuity, Security and Audit Plan** Analyse database risks and prepare a continuity plan covering transactions, security, backup, recovery, automation, and auditing.	-	10
Total		20	10

	Examination Style: 1. **Transaction Failure Diagnosis and Recovery Challenge (10 Marks):** • In this task, students will be given an unseen transaction-based database scenario containing multiple dependent SQL operations and possible failures such as incorrect updates, interrupted execution, partial completion, or concurrent access to the same record. Students are required to analyse the transaction flow, identify the point of failure, and apply suitable COMMIT, ROLLBACK, and SAVEPOINT operations to maintain correct data. Students must also identify possible concurrency problems such as lost update, dirty read, or unrepeatable read and explain how transaction control, isolation, and locking can preserve the ACID properties and reliability of the database. 2. **Security, Procedure and Trigger Automation (10 Marks):** • In this task, students will be given an unseen database requirement containing incorrectly assigned privileges, incomplete stored procedures, faulty triggers, or missing audit controls. Students are required to identify and correct the security and automation-related problems, create or modify database users, assign suitable privileges using GRANT, remove unnecessary privileges using REVOKE, reconstruct the stored procedure, and correct the trigger for automatic validation or audit-log maintenance. Students must execute valid and invalid operations to verify that user access, procedures, and triggers function according to the given business rules. 3. **ALA – Database Continuity, Security and Audit Plan (10 Marks):** • In this activity, students will select a real-world database application such as an academic record system, banking application, hospital management system, inventory system, library system, or online transaction platform. Students are required to study the possible transaction failures, data-loss risks, unauthorized-access risks, concurrency issues, and audit requirements of the selected system. They must prepare a database continuity and control plan defining suitable transaction-handling rules, user roles and privileges, locking requirements, backup schedule, recovery		

		procedure, stored-procedure requirements, trigger-based audit controls, and verification test cases. The final submission should include the selected system description, risk analysis, transaction-control plan, user-access matrix, backup and recovery plan, automation requirements, audit strategy, test-case table, and brief presentation submitted through the GMIU Web Portal.		

Suggested Specification table with Marks (Theory):100

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	10%	15%	30%	20%	10%	15%

Course Outcome:

After learning the course, the students should be able to:	
CO1	Explain DBMS fundamentals, architecture, and data models.
CO2	Design ER diagrams and relational schemas with constraints.
CO3	Apply SQL commands for database creation and data handling.
CO4	Develop advanced SQL queries and normalized database structures.
CO5	Implement transactions, security, recovery, procedures, and triggers.

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

From the content 10% topics are suggested for flipped mode instruction.

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory.

The internal evaluation will be done on the basis of Active Learning Assignment.

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

Reference Books:

[1] *Database System Concepts*, by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, McGraw Hill Education.
[2] *Fundamentals of Database Systems*, by Ramez Elmasri and Shamkant B. Navathe, Pearson Education.
[3] *Database Management Systems*, by Raghu Ramakrishnan and Johannes Gehrke, McGraw Hill Education.
[4] *An Introduction to Database Systems*, by C. J. Date, Pearson Education.
[5] *SQL, PL/SQL: The Programming Language of Oracle*, by Ivan Bayross, BPB Publications.

Suggested Rubrics:

Suggested Assessment Guidelines

Module-1: Fundamentals of Database Management Systems

DBMS Application Flow and Architecture Analysis (10 Marks)		
Criteria	**Description**	**Marks**
Data and User Analysis	Correct identification of data requirements, DBMS users, roles, and access requirements.	03
Application Data Flow	Clear representation of data entry, storage, retrieval, updating, and access flow.	03
Architecture Selection	Selection and justification of a suitable DBMS architecture for the given application.	04
	Total	10

Database Structure Evaluation and Implementation (10 Marks)		
Criteria	**Description**	**Marks**
Error Identification	Correct identification of unsuitable fields, data types, duplicated data, and missing identifiers.	03
Database Redesign	Clear and technically accurate presentation of differences or similarities between the given concepts.	03
SQL Implementation	Correct implementation and execution of insertion, retrieval, updating, and deletion operations.	04
	Total	10

Module-2: Database Design Using ER Model, Keys and Integrity Constraints

Requirement-to-ER Design Challenge (10 Marks)		
Criteria	**Description**	**Marks**
Requirement Analysis	Correct identification of data requirements, business rules, entities, and attributes.	03
ER Diagram Design	Accurate representation of relationships, cardinality, degree, and participation constraints.	04
Design Justification	Appropriate classification of attributes and justification of major design decisions.	03
	Total	10

Relational Schema Correction and Constraint Testing (10 Marks)		
Criteria	**Description**	**Marks**
Error Identification	Correct identification of missing keys, incorrect relationships, duplicated attributes, and invalid references.	03
Schema Correction	Proper conversion into relational tables with suitable keys, data types, and junction tables.	03
Constraint Implementation and Testing	Correct implementation and testing of keys and integrity constraints using valid and invalid records	04
	Total	10

Module-3: SQL Fundamentals, Data Manipulation and Report Generation

SQL Error Diagnosis and Query Reconstruction (10 Marks)		
Criteria	**Description**	**Marks**
SQL Diagnostic Analysis	Accurate analysis of syntax,[2] logical, constraint-related, and execution-sequence problems.	03
Query Reconstruction	Correct reconstruction of SQL commands according to the given requirements.	04
Execution and Output Verification	Successful execution, comparison of expected and actual outputs, and clear explanation of corrections.	03
	Total	10

Data Audit, Cleaning and Analytical Report Generation (10 Marks)		
Criteria	**Description**	**Marks**
Data-Quality Analysis	Correct identification and classification of incomplete, duplicate, invalid, and inconsistent records.	03
Data Cleaning	Appropriate correction or removal of incorrect records using suitable SQL commands.	03
Report Generation and Verification	Accurate generation and verification of filtered, sorted, aggregate, and grouped reports	04
	Total	10

Module-4: Advanced SQL, Query Optimization and Database Normalization

SQL Error Diagnosis and Query Reconstruction (10 Marks)		
Criteria	**Description**	**Marks**
Query and Relationship Analysis	Correct analysis of reporting requirements, table relationships, join conditions, and query logic.	03
Query Reconstruction	Appropriate use of joins, subqueries, nested queries, or set operations to generate accurate results.	04
Optimization and Output Verification	Improvement of query readability and efficiency with successful verification of the generated output	03
	Total	10

Normalization and Database Redesign Challenge (10 Marks)		
Criteria	**Description**	**Marks**
Dependency and Anomaly Analysis	Correct identification of keys, functional dependencies, redundancy, and database anomalies.	03
Normalization Process	Accurate decomposition of the given structure up to 3NF or BCNF.	03
Schema Redesign and Justification	Proper definition of primary and foreign keys with clear justification of the redesigned schema.	04
	Total	10

Module-5: Database Transactions, Concurrency Control, Security and Recovery

Transaction Failure Diagnosis and Recovery Challenge (10 Marks)		
Criteria	**Description**	**Marks**
Transaction Flow Analysis	Accurate analysis of dependent SQL operations, failure points, and transaction states.	03
Recovery Command Implementation	Correct use of COMMIT, ROLLBACK, and SAVEPOINT to maintain database consistency.	04
Concurrency and ACID Analysis	Appropriate analysis of concurrency problems, locking, isolation, and ACID properties.	03
	Total	10

Security, Procedure and Trigger Automation (10 Marks)		
Criteria	**Description**	**Marks**
User and Privilege Management	Correct identification of keys, functional dependencies, redundancy, and database anomalies.	03
Procedure and Trigger Reconstruction	Accurate correction and implementation of stored procedures and triggers according to business rules.	04
Security and Automation Verification	Successful testing of valid and invalid operations to verify access control and automated actions.	03
	Total	10