FACULTY OF ENGINEERING & TECHNOLOGY

Effective from Academic Batch: 2025-26

Programme: **Bachelor of Technology (Computer Science and Design)**

Semester: **III**

Course Code: **102040305**

Course Title: **Database Management Systems**

Course Group: **Professional Core Course**

Course Objectives:

This course aims to equip students with a comprehensive understanding of foundational database system concepts. Students will learn to design databases using Entity-Relationship (ER) and relational data models and implement them using SQL for precise data definition and manipulation. Furthermore, the curriculum explores critical backend processes, including transaction management, concurrency control, query optimization, and robust security protocols. Ultimately, this course prepares students to architect, deploy, and maintain highly efficient database systems for practical, real-world applications.

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
				Theory		J/V/P*		Total
Lecture	Tutorial	Practical		Internal	External	Internal	External	
3	0	2	4	50 / 18	50 / 18	25 / 9	25 / 9	150 / 54

* **J**: Jury; **V**: Viva; **P**: Practical

Detailed Syllabus:

Sr.	Contents	Hours
1	**Introduction to Database Systems:** Database system applications, purpose of database systems, database system concepts (data models, schemas, instances, views), database system architecture and components, data independence, classification of DBMS, challenges in DBMS, role of database administrator (DBA).	6

2	**Database Models:** Entity Relationship (ER) model: entities, attributes, keys, relationships, structural constraints, ER diagrams, design issues, weak entities, extended ER features (generalization, specialization, aggregation), ER to relational mapping. Relational data model: relations, schema-instance distinction, relational algebra (basic unary and binary operators).	**8**
3	**Relational Database Design & Normalization:** Importance of good schema design, functional dependencies, Armstrong's axioms, closure of attribute sets, minimal cover, normalization (1NF, 2NF, 3NF, BCNF), decomposition properties, multivalued dependencies and 4NF.	**8**
4	**SQL and PL/SQL:** Basics of SQL, DDL, DML, DCL commands, constraints (primary key, foreign key, unique, check, not null), aggregate functions, joins, subqueries, views, transaction control commands, introduction to PL/SQL, cursors, stored procedures, stored functions, triggers.	**7**
5	**Transaction Management and Recovery:** Transaction concepts, ACID properties, schedules and serializability, concurrency control issues, lock-based protocols, deadlock handling, recovery concepts, log-based recovery, checkpointing.	**6**
6	**Query Processing, Storage and Security:** Query processing steps, query optimization techniques, indexing and hashing, storage media and RAID, authentication and authorization, access control models (DAC, MAC, RBAC), database security basics	**5**
	Total:	**40**

List of Practicals / Tutorials:

1	**Creation of Database Objects and Integrity Constraints (DDL). Create database which consist of following table and insert appropriate data into it.** i. Table: regions • region_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT. • regionname: varchar(50), DEFAULT NULL. ii. Table: countries • country_id: char(2), Primary Key, NOT NULL. • countryname: varchar(40), NOT NULL. • region_id: int(11), Foreign Key (References regions.region_id).

iii. Table: locations

- location_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT.
- address: varchar(255), NOT NULL.
- postalcode: varchar(20), DEFAULT NULL.
- city: varchar(50), DEFAULT NULL.
- state: varchar(50), DEFAULT NULL.
- country_id: char(2), Foreign Key (References countries.country_id).

iv. Table: warehouses

- warehouse_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT.
- warehousename: varchar(255), DEFAULT NULL.
- location_id: int(11), Foreign Key (References locations.location_id).

v. Table: employees

- employee_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT.
- firstname: varchar(255), NOT NULL.
- lastname: varchar(255), NOT NULL.
- email: varchar(255), NOT NULL.
- phone: varchar(50), NOT NULL.
- hiredate: date, NOT NULL.
- manager_id: int(11), DEFAULT NULL (Self-referencing Foreign Key).
- jobtitle: varchar(255), NOT NULL.

vi. Table: product_categories

- category_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT.
- categoryname: varchar(255), NOT NULL.

vii. Table: products

- product_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT.
- productname: varchar(255), NOT NULL.
- description: varchar(2000), DEFAULT NULL.
- standardcost: int(11), DEFAULT NULL.
- listprice: int(11), DEFAULT NULL.
- category_id: int(11), NOT NULL, Foreign Key (References product_categories.category_id).

	viii. Table: customers • customer_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT. • name: varchar(255), NOT NULL. • address: varchar(255), DEFAULT NULL. • website: varchar(255), DEFAULT NULL. • creditlimit: int(11), DEFAULT NULL.
	ix. Table: orders • order_id: int(11), Primary Key, NOT NULL, AUTO_INCREMENT. • status: varchar(20), NOT NULL. • customer_id: int(11), DEFAULT NULL (Foreign Key referencing customers). • salesman_id: int(11), DEFAULT NULL (Foreign Key referencing employees). • order_date: date, NOT NULL. x. Table: order_items • order_id: int(11), Composite Primary Key, NOT NULL. • item_id: int(11), Composite Primary Key, NOT NULL. • product_id: int(11), NOT NULL (Foreign Key referencing products). • quantity: int(11), NOT NULL. • unit_price: int(11), NOT NULL. xi. Table: inventories • product_id: int(11), Composite Primary Key, NOT NULL. • warehouse_id: int(11), Composite Primary Key, NOT NULL. • quantity: int(11), NOT NULL.
2	**Table Cloning and Data Transformation** i. Create warehouse1 from the warehouses table without copying any existing data. ii. Create a full replica of the warehouses table, including all existing records. iii. Create table employee_m from employee using select clause having selected number of columns as employee_id, full_name. full_name column will be merging of first name and last name in columns. Apply constraint to last_column as NOT NULL.

3	**Schema Evolution and Maintenance** i. Add a salary column to employee_m with an integer data type and a NOT NULL constraint. ii. Modify the salary column to support decimal values using the float(8,2) type. iii. Rename the salary column to salary_e and change its type back to integer. iv. Add a date_of_joining column specifically positioned after the full_name column. v. Perform structural cleanup by dropping the date_of_joining column and renaming the table to employee_r vi. Remove the Primary Key constraint and then truncate all data from the employee_r table.
4	**Data Manipulation and Basic Retrieval (DML & DQL)** i. Retrieve and display all columns and rows from the products table. ii. Generate a list of unique (distinct) job titles held by employees. iii. Display the city and postalcode for all entries in the locations table.
5	**Advanced Filtering and Pattern Matching** i. List all employees whose last name contains the substring "ada". ii. Find employees whose last names start with "Jan" or end with the letters "na". iii. Search for employees with a 5-letter last name starting with "D" and having "a" as the third character. iv. Display customer names with a remark "high" if their credit limit is > 2500, else "low". v. Retrieve exactly the first 10 rows and then specifically the 4th row from the products table.
6	**Sorting, Restricting, and Set Operations (DRL)** i. Display all employee details sorted chronologically by their hiredate. ii. Sort the products table in descending order of listprice and then by category. iii. List countries belonging specifically to Africa, Asia, or America, arranged in that exact regional order. iv. Display name of products belongs to category 3, 1, 5 as per following. v. 3 P1, P2, P3 1 P7, P9 5 P5, P6 vi. List out products with minimum price from each category.

7	**Scalar and Aggregate Functions** i. Display the current system date with the column label "Date". ii. Calculate a 15% price increase for all products and display it as a whole number labeled "New Price". iii. Display capitalized names and their lengths for employees whose names start with J, A, or M. iv. Calculate the total years of employment for each employee based on their join date. v. Count the total number of employees for each unique job title.
8	**Complex Joins and Multi-Table Retrieval** i. List every product name alongside its category name, sorted alphabetically by category. ii. Display customer names and the total count of items they have ordered, highest volume first. iii. Find the names of customers who placed orders delivered from the "New Jersey" warehouse. iv. For every order, identify the single most expensive product and list it first. v. List the total number of products sold by each warehouse located in India, ordered alphabetically by warehouse.
9	**Transactions and Data Security** i. Perform a series of DML updates and use COMMIT to make them permanent. ii. Demonstrate the ROLLBACK command by reverting an accidental data deletion. iii. Create a database View that only shows product names and prices to restrict sensitive cost data. iv. Test the security of a view by attempting to drop it after creation. v. Identify the minimum product price within each category to support low-cost analysis
10	**PL/SQL Fundamentals and Control Structures** i. Write a basic PL/SQL block to insert a new category into the product_categories table. ii. Create a block using IF-THEN-ELSE to update a product's price based on its current value. iii. Implement a loop that prints the names of the top 5 customers by credit limit.
11	**Advanced PL/SQL (Cursors, Procedures, and Triggers)** i. Create a Cursor to iterate through and display employees hired in a specific year. ii. Develop a Stored Procedure that accepts a customer_id and returns their total order value. iii. Design a Database Trigger that logs an entry every time a new product is added.

12	**Mini-Project: System Implementation** i. Design an Entity-Relationship (ER) Diagram based on the provided technology hardware system. ii. Write the full script to create the schema with all required constraints and relationships. iii. Implement at least three complex queries used for monthly business reporting.

Reference Books:

1	A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th Edition, McGraw Hill Education.
2	Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Education.
3	C. J. Date, An Introduction to Database Systems, 8th Edition, Pearson Education.
4	Ivan Bayross, SQL, PL/SQL – The Programming Language of Oracle, BPB Publications.

Supplementary learning material:	
1	NPTEL Online Courses ● Database Management Systems – IIT Kharagpur ● Database Systems – IIT Bombay ● https://swayam.gov.in
2	Documentation, Tutorials, and Virtual Labs ● Virtual Labs: IIT Bombay VLABS https://vlabs.iitb.ac.in ● Official Documentation: MySQL Documentation, PostgreSQL Documentation https://dev.mysql.com/doc
3	Interactive Practice and Problem Solving ● SQL Training: SQLZoo, Mode SQL Tutorial https://sqlzoo.net ● Industry Problem Solving: HackerRank SQL Track, LeetCode Database Problems https://www.hackerrank.com/domains/sql ● Online Execution: Oracle Live SQL https://livesql.oracle.com
4	Real-World Datasets for Projects ● Kaggle Datasets: Kaggle Database & SQL Datasets ● Academic Repositories: UCI Machine Learning Repository

Pedagogy:

- Direct classroom teaching

- Audio Visual presentations/demonstrations
- Assignments/Quiz
- Continuous assessment
- Interactive methods
- Seminar/Poster Presentation
- Industrial/ Field visits
- Course Projects

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks in %						**R**: Remembering; **U**: Understanding; **A**: Applying; **N**: Analyzing; **E**: Evaluating; **C**: Creating
R	**U**	**A**	**N**	**E**	**C**	
20%	20%	20%	15%	15%	10%	

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.

Student Learning Outcomes (SLOs):

Sr.	Student Learning Outcome Statements	%weightage
SLO-1	Explain fundamental concepts, architecture, and components of database systems.	**15%**
SLO-2	Design database schemas using ER and relational data models.	**20%**
SLO-3	Apply normalization techniques to develop efficient and anomaly-free relational schemas.	**20%**
SLO-4	Implement SQL and PL/SQL programs for database definition and manipulation.	**25%**
SLO-5	Analyze transaction management, concurrency control, and recovery techniques.	**10%**
SLO-6	Evaluate query processing, storage structures, and database security mechanisms.	**10%**

Curriculum Revision:	
Version:	3.0
Drafted on (Month-Year):	March-2026
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	April-2029