INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE DESCRIPTOR

Course Title	**DATABASE MANAGEMENT SYSTEMS LABORATORY**					
Course Code	ACS104					
Programme	B.Tech					
Semester	III	IT				
	IV	CSE				
Course Type	Core					
Regulation	IARE - R16					
Course Structure	**Theory**			**Practical**		
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**	
	-	-	-	3	2	
Chief Coordinator	Dr. K. Suvarchala, Professor					
Course Faculty	Ms K. Radhika, Assistant Professor Ms K. Mayuri, Assistant professor Ms.E. Uma Shankari, Assistant Professor					

I. COURSE OVERVIEW:

This course introduces the core principles and techniques required in the design and implementation of database systems. This introductory application-oriented course covers the relational database systems RDBMS - the predominant system for business, scientific and engineering applications at present. It includes Entity-Relational model, Normalization, Relational model and data access queries as well as an introduction to SQL. It also provides students with theoretical knowledge and practical skills in the use of databases and database management systems in information technology applications.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites	Credits
UG	ACS102	II	Data Structures Lab	2

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Database Management Systems	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✔	Chalk & Talk	✘	Quiz	✘	Assignments	✘	MOOCs
✔	LCD / PPT	✘	Seminars	✘	Mini Project	✔	Videos
✔	Open Ended Experiments						

V. EVALUATION METHODOLOGY:

Each laboratory will be evaluated for a total of 100 marks consisting of 30 marks for internal assessment and 70 marks for semester end lab examination. Out of 30 marks of internal assessment, continuous lab assessment will be done for 20 marks for the day to day performance and 10 marks for the final internal lab assessment.

Semester End Examination (SEE): The semester end lab examination for 70 marks shall be conducted by two examiners, one of them being Internal Examiner and the other being External Examiner, both nominated by the Principal from the panel of experts recommended by Chairman, BOS.

The emphasis on the questions is broadly based on the following criteria:

20 %	To test the preparedness for the experiment.
20 %	To test the performance in the laboratory.
20 %	To test the calculations and graphs related to the concern experiment.
20 %	To test the results and the error analysis of the experiment.
20 %	To test the subject knowledge through viva – voce.

Continuous Internal Assessment (CIA):
CIA is conducted for a total of 30 marks (Table 1), with 20 marks for continuous lab assessment during day to day performance, 10 marks for final internal lab assessment.

Table 1: Assessment pattern for CIA

Component	Laboratory		Total Marks
Type of Assessment	Day to day performance	Final internal lab assessment	
CIA Marks	20	10	30

Continuous Internal Examination (CIE):
One CIE exams shall be conducted at the end of the 16th week of the semester. The CIE exam is conducted for 10 marks of 3 hours duration.

Preparation	Performance	Calculations and Graph	Results and Error Analysis	Viva	Total
2	2	2	2	2	10

VI. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Strength	Proficiency assessed by
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	Case study
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences	2	Lab exercise
PO 3	**Design / development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	3	Lab exercise, Case studies
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	2	Lab exercise
PO 12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	2	-

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Strength	Proficiency assessed by
PSO 1	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.	2	Case study
PSO 2	**Problem-Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.	3	Lab exercise, Case studies
PSO 3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.	3	-

3 = High; 2 = Medium; 1 = Low

VIII. COURSE OBJECTIVES (COs):

The course should enable the students to:	
I	Discuss the basic database concepts, applications, data models, schemas and instances.
II	Design Entity Relationship model for a database.
III	Demonstrate the use of constraints and relational algebra operations.
IV	Describe the basics of SQL and construct queries using SQL.
V	Understand the importance of normalization in databases.

IX. COURSE LEARNING OUTCOMES (CLOs):

CLO Code	CLO’s	At the end of the course, the student will have the ability to:	PO’s Mapped	Strength of Mapping
ACS104.01	CLO 1	Define the terminology, features and environment for database system	PO 1	2
ACS104.02	CLO 2	Implement database using DDL and DML commands with various constraints	PO 2, PO 3	2
ACS104.03	CLO 3	Define and illustrate the Relational Data Model with integrity constraints	PO 2, PO 3	2
ACS104.04	CLO 4	Use an SQL interface of a multi-user relational DBMS package to create users and grant permissions to secure database.	PO 3, PO 5	2
ACS104.05	CLO 5	Apply transaction control statements to recover database at various points	PO 2, PO 5	2
ACS104.06	CLO 6	Use SQL queries for data aggregation, calculations, views generation.	PO 2, PO 5	3
ACS104.07	CLO 7	Formulate broad range of correlated and nested queries by joining multiple relations	PO 2, PO 3, PO 5	3
ACS104.08	CLO 8	Demonstrate PL/SQL program including stored procedures, stored functions.	PO 3, PO 5	3
ACS104.09	CLO 9	Distinguish stored procedure from stored functions using SQL	PO 3, PO 5	3
ACS104.10	CLO 10	Demonstrate various triggers with constraints on relations	PO 3, PO 5	3
ACS104.11	CLO 11	Describe implantation of cursors to interact with application programs	PO 2, PO 3, PO 5	2
ACS104.12	CLO 12	Model the real world database systems using Entity Relationship Diagrams (ERD) from the requirements specification.	PO 3, PO 12	2
ACS104.13	CLO 13	Transform ER model into a relational database schema and implement schema using data SQL and/or utilities	PO 3, PO 12	2
ACS104.14	CLO 14	Apply normalization to relational schema through various normal form.	PO 3, PO 12	2
ACS104.15	CLO 15	Possess the knowledge and skills for employability and to succeed in national and international level competitive examinations.	PO 5, PO 12	3

3 = High; 2 = Medium; 1 = Low

X. MAPPING COURSE LEARNING OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

CLOs	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO 1	3												1		
CLO 2		3	1										2		

CLOs	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO 3		3	2										1	2	
CLO 4			2		3								2	2	
CLO 5		2			3								2	2	
CLO 6		2			3									3	
CLO 7		2	3		2									3	
CLO 8			3		2								2	3	
CLO 9			3		2								2	3	
CLO 10			3		2								2	3	
CLO 11		2	3		2								2	3	
CLO 12			3									2		2	
CLO 13			3									2		2	
CLO 14			3									2		2	
CLO 15					3							2		2	3

3 = High; 2 = Medium; 1 = Low

XI. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO 1, PO 2, PO 3, PO 5	SEE Exams	PO 1, PO 2, PO 3, PO 5	Assignments	-	Seminars	-
Laboratory Practices	PO 2, PO 3, PO 5	Student Viva	PO 2, PO 3	Mini Project	-	Certification	-
Term Paper	-						

XII. ASSESSMENT METHODOLOGIES - INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✘	Assessment of Mini Projects by Experts		

XIII. SYLLABUS

LIST OF EXPERIMENTS

WEEK-1	CREATION OF TABLES

1. Create a table called Employee with the following structure.

Name	Type
Empno	Number
Ename	Varchar2(20)

Job	Varchar2(20)
Mgr	Number
Sal	Number

a. Add a column commission with domain to the Employee table.
b. Insert any five records into the table.
c. Update the column details of job
d. Rename the column of Employ table using alter command.
e. Delete the employee whose empno is 19.

2. Create department table with the following structure.

Name	Type
Deptno	Number
Deptname	Varchar2(20)
location	Varchar2(20)

a. Add column designation to the department table.
b. Insert values into the table.
c. List the records of emp table grouped by deptno.
d. Update the record where deptno is 9.
e. Delete any column data from the table.

3. Create a table called Customer table

Name	Type
Cust name	Varchar2(20)
Cust street	Varchar2(20)
Cust city	Varchar2(20)

a. Insert records into the table.
b. Add salary column to the table.
c. Alter the table column domain.
d. Drop salary column of the customer table.
e. Delete the rows of customer table whose cust_city is ‘hyd’.

4. Create a table called branch table.

Name	Type
Branch name	Varchar2(20)
Branch city	Varchar2(20)
asserts	Number

a. Increase the size of data type for asserts to the branch.
b. Add and drop a column to the branch table.
c. Insert values to the table.
d. Update the branch name column
e. Delete any two columns from the table

5. Create a table called sailor table

Name	Type
Sid	Number
Sname	Varchar2(20)
rating	Varchar2(20)

a. Add column age to the sailor table.
b. Insert values into the sailor table.
c. Delete the row with rating >8.
d. Update the column details of sailor.
e. Insert null values into the table.

6. Create a table called reserves table

Name	Type
Boat id	Integer
sid	Integer
day	Integer

a. Insert values into the reserves table.
b. Add column time to the reserves table.
c. Alter the column day data type to date.
d. Drop the column time in the table.
e. Delete the row of the table with some condition.

WEEK -2 QUERIES USING DDL AND DML

1. a. Create a user and grant all permissions to the user.
   b. Insert the any three records in the employee table and use rollback. Check the result.
   c. Add primary key constraint and not null constraint to the employee table.
   d. Insert null values to the employee table and verify the result.
2. a. Create a user and grant all permissions to the user.
   b. Insert values in the department table and use commit.
   c. Add constraints like unique and not null to the department table.
   d. Insert repeated values and null values into the table.
3. a. Create a user and grant all permissions to the user.
   b. Insert values into the table and use commit.
   c. Delete any three records in the department table and use rollback.
   d. Add constraint primary key and foreign key to the table.
4. a. Create a user and grant all permissions to the user.
   b. Insert records in the sailor table and use commit.
   c. Add save point after insertion of records and verify save point.
   d. Add constraints not null and primary key to the sailor table.
5. a. Create a user and grant all permissions to the user.
   b. Use revoke command to remove user permissions.
   c. Change password of the user created.
   d. Add constraint foreign key and not null.
6. a. Create a user and grant all permissions to the user.
   b. Update the table reserves and use savepoint and rollback.
   c. Add constraint primary key , foreign key and not null to the reserves table

d. Delete constraint not null to the table column.

WEEK -3 QUERIES USING AGGREGATE FUNCTIONS

1. a. By using the group by clause, display the enames who belongs to deptno 10 along with average salary.
   b. Display lowest paid employee details under each department.
   c. Display number of employees working in each department and their department number.
   d. Using built in functions, display number of employees working in each department and their department name from dept table. Insert deptname to dept table and insert deptname for each row, do the required thing specified above.
   e. List all employees which start with either B or C.
   f. Display only these ename of employees where the maximum salary is greater than or equal to 5000.
2. a. Calculate the average salary for each different job.
   b. Show the average salary of each job excluding manager.
   c. Show the average salary for all departments employing more than three people.
   d. Display employees who earn more than the lowest salary in department 30
   e. Show that value returned by sign (n) function.
   f. How many days between day of birth to current date.
3. a. Show that two substring as single string.
   b. List all employee names, salary and 15% rise in salary.
   c. Display lowest paid emp details under each manager
   d. Display the average monthly salary bill for each deptno.
   e. Show the average salary for all departments employing more than two people.
   f. By using the group by clause, display the eid who belongs to deptno 05 along with average salary.
4. a. Count the number of employees in department 20
   b. Find the minimum salary earned by clerk.
   c. Find minimum, maximum, average salary of all employees.
   d. List the minimum and maximum salaries for each job type.
   e. List the employee names in descending order.
   f. List the employee id, names in ascending order by empid.
5. a. Find the sids ,names of sailors who have reserved all boats called “INTERLAKE Find the age of youngest sailor who is eligible to vote for each rating level with at least two such sailors.
   b. Find the sname , bid and reservation date for each reservation.
   c. Find the ages of sailors whose name begin and end with B and has at least 3 characters.
   d. List in alphabetic order all sailors who have reserved red boat.
   e. Find the age of youngest sailor for each rating level.
6. a. List the Vendors who have delivered products within 6 months from order date.
   b. Display the Vendor details who have supplied both Assembled and Sub parts.
   c. Display the Sub parts by grouping the Vendor type (Local or Non Local).
   d. Display the Vendor details in ascending order.
   e. Display the Sub part which costs more than any of the Assembled parts.
   f. Display the second maximum cost Assembled part.

WEEK - 4 PROGRAMS ON PL/SQL

1. a. Write a PL/SQL program to swap two numbers.
   b. Write a PL/SQL program to find the largest of three numbers.
2. a. Write a PL/SQL program to find the total and average of 6 subjects and display the grade.
   b. Write a PL/SQL program to find the sum of digits in a given number.
3. a. Write a PL/SQL program to display the number in reverse order.
   b. Write a PL / SQL program to check whether the given number is prime or not.
4. a. Write a PL/SQL program to find the factorial of a given number.
   b. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius and area.
5. a. Write a PL/SQL program to accept a string and remove the vowels from the string. (When ‘hello’ passed to the program it should display ‘Hll’ removing e and o from the world Hello).
   b. Write a PL/SQL program to accept a number and a divisor. Make sure the divisor is less than or equal to 10. Else display an error message. Otherwise Display the remainder in words.

WEEK -5 PROCEDURES AND FUNCTIONS

1. Write a function to accept employee number as parameter and return Basic +HRA together as single column.
2. Accept year as parameter and write a Function to return the total net salary spent for a given year.
3. Create a function to find the factorial of a given number and hence find NCR.
4. Write a PL/SQL block o pint prime Fibonacci series using local functions.
5. Create a procedure to find the lucky number of a given birth date.
6. Create function to the reverse of given number.

WEEK-6 TRIGGERS

1. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:
   CUSTOMERS table:

ID	NAME	AGE	ADDRESS	SALARY
1	Alive	24	Khammam	2000
2	Bob	27	Kadappa	3000
3	Catri	25	Guntur	4000
4	Dena	28	Hyderabad	5000
5	Eeshwar	27	Kurnool	6000
6	Farooq	28	Nellur	7000

2. Creation of insert trigger, delete trigger, update trigger practice triggers using the passenger database.
   Passenger(Passport_ id INTEGER PRIMARY KEY, Name VARCHAR (50) Not NULL,

Age Integer Not NULL, Sex Char, Address VARCHAR (50) Not NULL);

a. Write a Insert Trigger to check the Passport_id is exactly six digits or not.
b. Write a trigger on passenger to display messages ‘1 Record is inserted’, ‘1 record is deleted’, ‘1 record is updated’ when insertion, deletion and updation are done on passenger respectively.

3. Insert row in employee table using Triggers. Every trigger is created with name any trigger have same name must be replaced by new name. These triggers can raised before insert, update or delete rows on data base. The main difference between a trigger and a stored procedure is that the former is attached to a table and is only fired when an INSERT, UPDATE or DELETE occurs.
4. Convert employee name into uppercase whenever an employee record is inserted or updated. Trigger to fire before the insert or update.
5. Trigger before deleting a record from emp table. Trigger will insert the row to be deleted into table called delete _emp and also record user who has deleted the record and date and time of delete.
6. Create a transparent audit system for a table CUST_MSTR. The system must keep track of the records that are being deleted or updated.

WEEK-7 PROCEDURES

1. Create the procedure for palindrome of given number.
2. Create the procedure for GCD: Program should load two registers with two Numbers and then apply the logic for GCD of two numbers. GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisors of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD is found.
3. Write the PL/SQL programs to create the procedure for factorial of given number.
4. Write the PL/SQL programs to create the procedure to find sum of N natural number.
5. Write the PL/SQL programs to create the procedure to find Fibonacci series.
6. Write the PL/SQL programs to create the procedure to check the given number is perfect or not.

WEEK-8 CURSORS

1. Write a PL/SQL block that will display the name, dept no, salary of fist highest paid employees**.**
2. Update the balance stock in the item master table each time a transaction takes place in the item transaction table. The change in item master table depends on the item id is already present in the item master then update operation is performed to decrease the balance stock by the quantity specified in the item transaction in case the item id is not present in the item master table then the record is inserted in the item master table.
3. Write a PL/SQL block that will display the employee details along with salary using cursors.
4. To write a Cursor to display the list of employees who are working as a Managers or Analyst.
5. To write a Cursor to find employee with given job and deptno.
6. Write a PL/SQL block using implicit cursor that will display message, the salaries of all the employees in the ‘employee’ table are updated. If none of the employee’s salary are updated we get a message 'None of the salaries were updated'. Else we get a message

like for example, 'Salaries for 1000 employees are updated' if there are 1000 rows in ‘employee’ table.

WEEK-9 CASE STUDY: BOOK PUBLISHING COMPANY

A publishing company produces scientific books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more publications.

A publication covers essentially one of the specialist subjects and is normally written by a single author. When writing a particular book, each author works with on editor, but may submit another work for publication to be supervised by other editors. To improve their competitiveness, the company tries to employ a variety of authors, more than one author being a specialist in a particular subject for the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams.

WEEK -10 CASE STUDY GENERAL HOSPITAL

A General Hospital consists of a number of specialized wards (such as Maternity, Pediatric, Oncology, etc). Each ward hosts a number of patients, who were admitted on the recommendation of their own GP and confirmed by a consultant employed by the Hospital. On admission, the personal details of every patient are recorded. A separate register is to be held to store the information of the tests undertaken and the results of a prescribed treatment. A number of tests may be conducted for each patient. Each patient is assigned to one leading consultant but may be examined by another doctor, if required. Doctors are specialists in some branch of medicine and may be leading consultants for a number of patients, not necessarily from the same ward. For the above case study, do the following.

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams.

WEEK -11 CASE STUDY: CAR RENTAL COMPANY

A database is to be designed for a car rental company. The information required includes a description of cars, subcontractors (i.e. garages), company expenditures, company revenues and customers. Cars are to be described by such data as: make, model, year of production, engine size, fuel type, number of passengers, registration number, purchase price, purchase date, rent price and insurance details. It is the company policy not to keep any car for a period exceeding one year. All major repairs and maintenance are done by subcontractors (i.e. franchised garages), with whom CRC has long-term agreements. Therefore the data about garages to be kept in the database includes garage names, addresses, range of services and the like. Some garages require payments immediately after a repair has been made; with others CRC has made arrangements for credit facilities. Company expenditures are to be registered for all outgoings connected with purchases, repairs, maintenance, insurance etc. Similarly the cash inflow coming from all sources: Car hire, car sales, insurance claims must be kept of file. CRC maintains a reasonably stable client base. For this privileged category of customers special credit card facilities are provided. These customers may also book in advance a particular car. These reservations can be made for any period of time up to one month. Casual customers must pay a deposit for an estimated time of rental, unless they wish to pay by credit card. All major credit cards are accepted. Personal details such as name, address, telephone number, driving license, number about each customer are kept in the database. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

   Create the logical data model using E-R diagrams.

WEEK-12 CASE STUDY: STUDENT PROGRESS MONITORING SYSTEM

A database is to be designed for a college to monitor students' progress throughout their course of study. The students are reading for a degree (such as BA, BA (Hons) M.Sc., etc) within the framework of the modular system. The college provides a number of modules, each being characterized by its code, title, credit value, module leader, teaching staff and the department they come from. A module is coordinated by a module leader who shares teaching duties with one or more lecturers. A lecturer may teach (and be a module leader for) more than one module. Students are free to choose any module they wish but the following rules must be observed: Some modules require pre- requisites modules and some degree programmes have compulsory modules. The database is also to contain some information about students including their numbers, names, addresses, degrees they read for, and their past performance i.e. modules taken and examination results. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.
3. Create the logical data model i.e., ER diagrams.
4. Comprehend the data given in the case study by creating respective tables with primary keys and foreign keys wherever required.
5. Insert values into the tables created (Be vigilant about Master- Slave tables).
6. Display the Students who have taken M.Sc course.
7. Display the Module code and Number of Modules taught by each Lecturer.
8. Retrieve the Lecturer names who are not Module Leaders.
9. Display the Department name which offers ‘English’ module.
10. Retrieve the Prerequisite Courses offered by every Department (with Department names).
11. Present the Lecturer ID and Name who teaches ‘Mathematics’.
12. Discover the number of years a Module is taught.
13. List out all the Faculties who work for ‘Statistics’ Department.
14. List out the number of Modules taught by each Module Leader.
15. List out the number of Modules taught by a particular Lecturer.
16. Create a view which contains the fields of both Department and Module tables. (Hint- The fields like Module code, title, credit, Department code and its name).
17. Update the credits of all the prerequisite courses to 5. Delete the Module ‘History’ from the Module table.

Text Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill, 4th Edition, 2002.
2. Ivan Bayross, “SQL, PL/SQL The programming language of oracle”, BPB publications, 4th Revised Edition, 2010.

Reference Books:

1. Ramez Elmasri, Shamkant, B. Navathe, “Database Systems”, Pearson Education, 6th Edition, 2013.
2. Peter Rob, Carles Coronel, “Database System Concepts”, Cengage Learning, 7th Edition, 2008.
3. M L Gillenson, “Introduction to Database Management”, Wiley Student Edition, 2012.

Web References:
1. https://www.tutorialspoint.com/dbms/ 2. https://www.javatpoint.com/dbms-tutorial

XIV. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Week No.	Topics to be covered	CLOs	Reference
1.	Introduction to database management system environments	CLO 1	T1:4.1, T2:1.1
2.	Creation of tables using DDL and DML commands	CLO 2	T1:4.9,4.11, T2:7
3.	Working with integrity constraints	CLO 3	T1:3,T2:8
4.	Working with DCL and TCL commands	CLO 4, CLO 5	T1:6.6,T2:12
5.	Queries using aggregate functions	CLO 6	T1:4.4,T2:10
6.	Nested queries using comparision keywords and logical operators	CLO 7	T1:4.6, T2:10
7.	Working with Programs on pl/sql	CLO 8, CLO 9	T2:15
8.	Working with Procedures	CLO 8, CLO 9	T2:18
9.	Working with Triggers	CLO 10	T2:18
10.	Working with functions	CLO 8, CLO 9	T2:18
11.	Working with Cursors	CLO 11	T2:10
12.	Case study: Book Publishing Company	CLO 12, CLO 13, CLO 14	T1:2,T2:1
13.	Case study General Hospital	CLO 12, CLO 13, CLO 14	T1:2, T2:1
14.	Case study: Car Rental Company	CLO 12, CLO 13, CLO 14	T1:2, T2:1
15.	Case study: Student Progress Monitoring System	CLO 12, CLO 13, CLO 14	T1:2, T2:1

XV. GAPS IN THE SYLLABUS - TO MEET INDUSTRY / PROFESSION REQUIREMENTS:

S No.	Description	Proposed action	Relevance with POs	Relevance with PSOs
1	Implementation of views using SQL.	Lab experiments	PO 2	PSO 2
2	Practical Implementation of assertions using PL/SQL	Lab experiments	PO 2	PSO 2
3	Encourage students to solve real time applications and prepare towards competitive examinations.	Lab experiments	PO 2, PO 12	PSO 2

Prepared by:
Dr. K. Suvarchala, Professor

HOD, CSE