DATABASE MANAGEMENT SYSTEMS LABORATORY MANUAL (CS522L5C)

V Semester B.E.

(Academic Year: 2024-25)

Prepared By:

Mr. Vasudeva Rao P V

Ms. Sneha Bose

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575007

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution

Institute Vision

To be a premier institution in Technology and Management by fostering excellence in education, innovation, incubation and values to inspire and empower the young minds.

Institute Mission

- Creating an academic ambience to impart holistic education focusing on individual growth, integrity, ethical values and social responsibility.
- Develop skill based learning through industry-institution interaction to enhance competency and promote entrepreneurship.
- Fostering innovation and creativity through competitive environment with state-of-the-art infrastructure.

Department Vision

To be a center of excellence in Information Science and Engineering through modern teaching pedagogy, innovation and incubation.

Department Mission

- Creating competitive ambience to enhance the innovative and experiential learning process through state-of the art infrastructure.
- Grooming young minds through industry-institute interactions to solve societal issues and inculcate affinity towards research and entrepreneurship.
- Promoting team work and leadership qualities through inter-disciplinary activities in diversified

  areas of information science and engineering.

Program Educational Objectives (PEOs):

PEO1: Possess theoretical and practical knowledge to identify, scrutinize, formulate and solve challenging problems related to dynamically evolving data science.

PEO2: Inculcate core competency, professionalism and ethics to cater industrial needs and to solve societal problems.

PEO3: Engage in Lifelong learning and stay intact to the transformation in technologies and pursue research.

Program Outcomes (POs) and Program Specific Outcomes (PSOs)

PO1. **Engineering Knowledge**: Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem Analysis**: Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3. **Design/ Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal and environmental considerations.

PO4. **Conduct Investigations** of complex problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.

PO5. **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.

PO6. **The Engineer and Society**: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7. **Environment and Sustainability**: Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

PO8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

PO9. **Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

PO11. **Project Management and Finance**: Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. **Life-long Learning**: Recognize the need for and have the preparation and ability to Engage in independent

PSO1: Able to Exhibit competency and skills in distributed computing, information security, cyber security, data analytics and machine learning.

PSO2: Equip student with skills to provide sustainable solutions for implementing and testing information science projects.

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575007

DATABASE MANAGEMENT SYSTEMS LABORATORY

(effective from the Academic Year 2023 - 24)

V SEMESTER

Course Code	**CS522L5C**	CIA Marks	50
Number of Contact Hours/Week (L: T: P: S)	0:0:2:0	SEE Marks	50
Total Hours of Pedagogy	**20P**	Exam Hours	03

CREDITS – 1

PREREQUISITES:

- Basics of SQL

COURSE OBJECTIVES:

- Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers.
- Strong practice in SQL programming through a variety of database problems.
- Develop database applications using front-end tools and back-end DBMS.

TEACHING - LEARNING STRATEGY:

Following are some sample strategies that can be incorporate for the Course Delivery

- Chalk and Talk Method/Blended mode method
- Power point Presentation
- Expert talk/Webinar/Seminar
- Video streaming/Self-Study/Simulations
- Peer-to-Peer activities
- Activity/Problem-based learning
- Case studies
- MOOC/NPTEL courses
- Any other innovative initiatives with respect to the course contents

LIST OF EXPERIMENTS

Sl. No.	Description
	Part-A
1	Consider the following schema for Bank database: **BRANCH(Branch_id, Bank_name, Branch_name, Assets)** **ACCOUNT(Acc_no, Branch_id, Account_Type, Account_Balance, Customer_id)** **CUSTOMER(Customer_id, Customer_name, Customer_age, Customer_address, Customer_phone)** **LOAN(Loan_number, Branch_id, Amount, Customer_id)** Note: Account_Type may be of following: Savings, Recurrent, Fixed Deposit Write SQL queries to 1. Find all the customers who have at least one account at the “Mangaluru” branch. 2. Find names of the depositors who have deposited highest amount among all the customers. 3. Retrieve the customer name and loan amount of a customer who borrowed a loan more than Rs. 5,00,000. 4. Retrieve the details of bank branch with maximum and minimum assets among the various branches.

	5. Demonstrate how you delete all account tuples at every branch located in a specific city.
2	Consider the following schema for a Library database: **BOOK(Book_id, Title, Publisher_Name, Pub_Year)** **BOOK_AUTHORS(Book_id, Author_Name)** **PUBLISHER(Name, Address, Phone)** **BOOK_COPIES(Book_id, Programme_id, No-of_Copies)** **BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date)** **LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address)** Write SQL queries to 1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each programme, etc. 2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2023 to Jun 2023. 3. Delete a book in the BOOK table. Update the contents of other tables to reflect this data manipulation operation. 4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query. 5. Create a view of all books and its number of copies that are currently available in the Library
3	Consider the schema for College database: **STUDENT(USN, SName, Address, Phone, Gender)** **SEMSEC(SSID, Sem, Sec)** **CLASS(USN, SSID)** **COURSE(Subcode, Title, Sem, Credits)** **IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)** Write SQL queries to 1. List all the student details studying in the fourth semester “C‟ section. 2. Compute the total number of male and female students in each semester and in each section. 3. Create a view of Test1 marks of student USN “4SF20CD001” in all courses. 4. Calculate the FinalIA (average of three test marks) and update the corresponding table for all students. 5. Categorize students based on the following criterion: If FinalIA = 45 to 50 then CAT = “Outstanding” If FinalIA= 40 to 45 then CAT= “Good” If FinalIA = 30 to 40 then CAT = “Average” If FinalIA< 30 then CAT = “Weak” Give these details only for 8th semester A, B, and C section students.
4	Consider the schema for Company database: **EMPLOYEE (Eid, Name, Address, Gender, Salary, SuperEid, Dno)** **DEPARTMENT (Dnum, Dname, DMgr_id, Mgr_start_date)** **DLOCATION (Dno, Dlocation)** **PROJECT (Pnum, Pname, Plocation, Dno)** **WORKS_ON (Eid, Pno, Hours)**

DEPENDENT (Empid, Dep_name, Gender, Bdate, Relationship)

Write SQL queries to

1. Make a list of all project numbers for projects that involve an employee whose name is “Rahul”, either as a worker or as a manager of the department that controls the project.
2. Show the resulting salaries if every employee working on the “IoT” project is given a 10 percent raise.
3. Find the sum of the salaries of all employees of the “Accounts” department, as well as the maximum salary, the minimum salary, and the average salary in this department.
4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).
5. Create a view Dept_info that gives details of department name, number of employees and total salary of each department.

5

Consider the schema for Airline database:

Flights (Flight_num, Source, Destination, Distance, Departs, Arrives, Price)

Aircraft (Aid, Aname, Cruising_range)

Certified (Emp_id, Aid)

Employees (Emp_id, Ename, Salary)

Note: The Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write SQL queries to

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
2. For each pilot who is certified for more than three aircrafts, find the Emp_id and the maximum cruising range of the aircraft for which she or he is certified.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Mumbai.
4. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
5. Find the employee name and salary earning the second highest salary.

Part-B: Mini Project

- For any societal problem statement selected.
- Make sure that the application should have five or more tables, one trigger and one Stored procedure.
- Mobile applications are strictly prohibited.
- The mini project team may consist of a maximum of two members.
- The areas for problem statement may include, but not limited to the following :
  - ❖ Educational sector
  - ❖ Environmental issues
  - ❖ Healthcare
  - ❖ Women empowerment
  - ❖ Child care
  - ❖ Banking sector

COURSE OUTCOMES		
Upon completion of this course, the students will be able to:		
CO No.	**Course Outcome Description**	**Bloom's Taxonomy Level**
CO1	Use SQL programming and different concepts of DBMS to create, update and query on the Bank and Library databases.	CL3
CO2	Demonstrate SQL programming and different concepts of DBMS to create, update and query on the College database.	CL3
CO3	Illustrate the concepts of SQL programming and DBMS to create, update and query on the Company database.	CL3
CO4	Create, update and query on the Airline database by using different concepts of DBMS and SQL programming.	CL3
CO5	Design, implement and demonstrate a database application using front end tools and Compile the working with well document using modern tool.	CL6

CO-PO-PSO MAPPING

CO No.	**Programme Outcomes (PO)**												**Programme Specific Outcomes (PSO)**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	3	3	3	2	2			1				2	3	2
CO2	3	3	3	2	2			1				2	3	2
CO3	3	3	3	2	2			1				2	3	2
CO4	3	3	3	2	2			1				2	3	2
CO5	3	3	3	2	2			1				2	3	2
CO6	3	3	3	3	3	2		3	3	3	3	2	3	2
3: Substantial (High)				**2: Moderate (Medium)**					**1: Poor (Low)**					

ASSESSMENT STRATEGY

Assessment will be both CIA and SEE. Students learning will be assessed using Direct and Indirect methods:

Sl. No.	**Assessment Description**	**Weightage (%)**	**Max. Marks**
1	**Continuous Internal Assessment (CIA)**	**100 %**	**50**
	Laboratory Work (A)	50 %	25
	Laboratory Test (B)	30 %	15
	Open Ended Experiments /Mini Projects (C)	20 %	10
2	**Semester End Examination (SEE)**	**100 %**	**50**

List of Experiments

EXPERIMENT No. 01: Bank Database

1.1 Experiment Details:

Consider the following schema for Bank database:

BRANCH(Branch_id,Bank_name,Branch_name,Assets)

ACCOUNT(Acc_no,Branch_id,Account_Type,Account_Balance,Customer_id)

CUSTOMER(Customer_id,Customer_name,Customer_age,Customer_address,Customer_phone)

LOAN(Loan_number,Branch_id,Amount,Customer_id)

Note: Account_Type may be of following: Savings, Recurrent, Fixed Deposit

Write SQL queries to

1. Find all the customers who have at least one account at the "Mangaluru" branch.
2. Find names of the depositors who have deposited highest amount among all the customers.
3. Retrieve the customer name and loan amount of a customer who borrowed a loan more than Rs.5,00,000.
4. Retrieve the details of bank branch with maximum and minimum assets among the various branches.
5. Demonstrate how you delete all account tuples at every branch located in a specific city.

1.2 Software Required:

Operating System: Windows 7

Software: Oracle 10g (SQL Plus)

1.3 Pre-Requisite:

Basics of DBMS, SQL Programming

1.4 Procedure:

TABLE CREATION:

```
CREATE TABLE BRANCH(
BRANCH_ID VARCHAR(10),
BANK_NAME VARCHAR(15),
BRANCH_NAME VARCHAR(20),
ASSETS INT NOT NULL,
PRIMARY KEY(BRANCH_ID) );

CREATE TABLE CUSTOMER(
CUSTOMER_ID VARCHAR(10),
CUSTOMER_NAME VARCHAR(20),
CUSTOMER_AGE INT,
CUSTOMER_ADDRESS VARCHAR(20),
CUSTOMER_PHONE INT,
PRIMARY KEY(CUSTOMER_ID) );

CREATE TABLE ACCOUNT(
ACC_NO INT,
BRANCH_ID VARCHAR(10),
ACCOUNT_TYPE VARCHAR(10),
ACCOUNT_BALANCE INT,
CUSTOMER_ID VARCHAR(10),
PRIMARY KEY(ACC_NO),
FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID) ON DELETE CASCADE,
FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID) ON DELETE CASCADE );
```

```
CREATE TABLE LOAN(

LOAN_NUMBER VARCHAR2(5),

BRANCH_ID VARCHAR(10),

AMOUNT INT,

CUSTOMER_ID VARCHAR(10),

PRIMARY KEY(LOAN_NUMBER),

FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID) ON   DELETE
CASCADE,

FOREIGN KEY(CUSTOMER_ID) REFERENCESCUSTOMER(CUSTOMER_ID) ON DELETE
CASCADE);


COMMIT;
```

Commit complete.

INSERTION OF VALUES:

```
SQL> INSERT INTO BRANCH VALUES
('&BRANCH_ID','&BANK_NAME','&BRANCH_NAME','&ASSETS');

Enter value for branch_id: B1

Enter value for bank_name: CANARA

Enter value for branch_name: MANGALURU

Enter value for assets: 60000000

old   1: INSERT INTO BRANCH VALUES

        ('&BRANCH_ID','&BANK_NAME','&BRANCH_NAME','&ASSETS')

new   1: INSERT INTO BRANCH VALUES('B1','CANARA','MANGALURU','60000000')

SQL> SELECT * FROM BRANCH;

  BRANCH_ID        BANK_NAME          BRANCH_NAME          ASSETS
  ----------    ---------------   --------------------   ----------
     B1             CANARA            MANGALURU          60000000
     B2         BANK OF BARODA        MANGALURU          70000000
     B3             CANARA            KASARAGOD          50000000
     B4              SBI              BENGALURU          30000000
     B5          UNION BANK             DELHI            20000000


SQL> INSERT INTO CUSTOMER VALUES
('&CUSTOMER_ID','&CUSTOMER_NAME','&CUSTOMER_AGE','&CUSTOMER_ADDRESS'
,'&CUSTOMER_PHONE');

Enter value for customer_id: C1

Enter value for customer_name: RAVI

Enter value for customer_age: 22

Enter value for customer_address: MANGALURU

Enter value for customer_phone: 8745263258

old   1: INSERT INTO CUSTOMER VALUES('&
CUSTOMER_ID','&CUSTOMER_NAME','&CUSTOMER_AGE','&CUSTOMER_ADDRESS','&
CUSTOMER_PHONE');

new   1: INSERT INTO CUSTOMER VALUES('C1','RAVI','22','MANGALURU','8745263258')
```

```
SQL> SELECT * FROM CUSTOMER;

CUSTOMER_ID   CUSTOMER_NAME   CUSTOMER_AGE   CUSTOMER_ADDRESS   CUSTOMER_PHONE
----------    --------------------   ------------   --------------------   --------------
    C1            RAVI             22          MANGALURU        8745263258
    C2            ASHA             26            DELHI          9874564123
    C3            VARUN            23          KASRAGOD         7895462325
    C4            ARPITHA          22          MANGALURU        9856325123
    C5            SACHIN           23          BENGALORE        7854136548

SQL> INSERT INTO ACCOUNT VALUES
('&ACC_NO','&BRANCH_ID','&ACCOUNT_TYPE','&ACCOUNT_BALANCE','&CUSTOMER_
ID');
Enter value for acc_no: 123
Enter value for branch_id: B1
Enter value for account_type: SAVINGS
Enter value for account_balance: 10000
Enter value for customer_id: C1
old   1: INSERT INTO ACCOUNT VALUES
         ('&ACC_NO','&BRANCH_ID','&ACCOUNT_TYPE','&ACCOUNT_BALANCE','&CUST
         OMER_ID')
new   1: INSERT INTO ACCOUNT VALUES('123','B1','SAVINGS','10000','C1')


SQL> SELECT * FROM ACCOUNT;

                                            ACCOUNT_BAL
ACC_NO     BRANCH_ID     ACCOUNT_TYPE           ANCE        CUSTOMER_ID
----------  ----------     ----------       ---------------    ----------
  123          B1          SAVINGS            10000            C1
  456          B5          RECURRING          20000            C2
  789          B1          SAVINGS            30000            C1
  1122         B2          FD                 5000             C3
  1334         B1          SAVINGS            10000            C4
  1234         B3          FD                 90000            C5
  5876         B4          RECURRING          80000            C3
```

```
SQL>  INSERT INTO LOAN
VALUES('&LOAN_NUMBER','&BRANCH_ID','&AMOUNT','&CUSTOMER_ID');

Enter value for loan_number: L1

Enter value for branch_id: B1

Enter value for amount: 500000

Enter value for customer_id: C1

old   1:  INSERT INTO LOAN
          VALUES('&LOAN_NUMBER','&BRANCH_ID','&AMOUNT','&CUSTOMER_ID')

new   1:  INSERT INTO LOAN VALUES('L1','B1','500000','C1')


SQL> SELECT* FROM LOAN;

          LOAN     BRANCH_ID       AMOUNT       CUSTOMER_ID
          -----    ----------      ---------    ----------
          L1       B1              500000       C1
          L2       B2              50000        C2
          L3       B3              40000        C3
          L4       B2              565000       C4
          L5       B4              955000       C5
          L6       B5              20000        C2
```

1.5 RESULTS

QUERY 1:

```
SQL>SELECT C.CUSTOMER_ID,C.CUSTOMER_NAME
FROM CUSTOMER C,ACCOUNT A,BRANCH B
WHERE B.BRANCH_NAME='MANGALURU'AND B.BRANCH_ID=A.BRANCH_ID
AND A.CUSTOMER_ID=C.CUSTOMER_ID;
```

```
CUSTOMER_ID     CUSTOMER_NAME
 ----------     --------------------
    C1               RAVI
    C1               RAVI
    C3               VARUN
    C4               ARPITHA
```

QUERY 2:

```
SQL>SELECTC.CUSTOMER_ID,C.CUSTOMER_NAME,A.ACCOUNT_BALANCE
FROM CUSTOMER C,ACCOUNT A
WHERE C.CUSTOMER_ID=A.CUSTOMER_ID AND
 ACCOUNT_BALANCE=(SELECT MAX(ACCOUNT_BALANCE)FROM ACCOUNT);
```

```
CUSTOMER_ID     CUSTOMER_NAME       ACCOUNT_BALANCE
 ----------     --------------------   ---------------
    C5              SACHIN               90000
```

QUERY 3:

```
SQL>SELECT C.CUSTOMER_NAME ,L.AMOUNT
FROM CUSTOMER C,LOAN L
WHERE C.CUSTOMER_ID=L.CUSTOMER_ID AND AMOUNT>500000;
```

```
CUSTOMER_NAME          AMOUNT
--------------------   ----------
   ARPITHA             565000
   SACHIN              955000
```

QUERY 4:

```
SQL>SELECT BANK_NAME,BRANCH_NAME,ASSETS
FROM BRANCH
WHERE ASSETS=(SELECT MAX(ASSETS)FROM BRANCH)
UNION
SELECT BANK_NAME,BRANCH_NAME,ASSETS
FROM BRANCH
WHERE ASSETS=(SELECT MIN(ASSETS)FROM BRANCH);

     BANK_NAME          BRANCH_NAME          ASSETS
     ----------       ---------------   --------------------

  BANK OF BARODA        MANGALURU           70000000
     UNION BANK            DELHI            20000000
```

QUERY 5:

```
SQL> SELECT * FROM BRANCH;

BRANCH_ID        BANK_NAME          BRANCH_NAME         ASSETS
----------    ---------------   --------------------   ----------
   B1             CANARA            MANGALURU          60000000
   B2         BANK OF BARODA        MANGALURU          70000000
   B3             CANARA            KASARAGOD          50000000
   B4              SBI              BENGALURU          30000000
   B5           UNION BANK            DELHI            20000000

SQL> DELETE FROM BRANCH WHERE BRANCH_NAME='DELHI';
   1 row deleted.
SQL> SELECT * FROM BRANCH;

BRANCH_ID        BANK_NAME          BRANCH_NAME         ASSETS
----------    ---------------   --------------------   ----------
   B1             CANARA            MANGALURU          60000000
   B2         BANK OF BARODA        MANGALURU          70000000
   B3             CANARA            KASARAGOD          50000000
   B4              SBI              BENGALURU          30000000
```

1.6 Pre – Experiment Questions:

1. What is database?
2. What is DBMS?
3. What is a Database system?
4. Advantages of DBMS?

1.7 Post – Experiment Questions:

1. Define the "integrity rules"
2. What is extension and intension?
3. What is Data Independence?
4. What is Data Model?

EXPERIMENT No. 02: Library Database

2.1 Experiment Details:

Consider the following schema for a Library database:

BOOK(Book_id,Title,Publisher_Name,Pub_Year)

BOOK_AUTHORS(Book_id, Author_Name)

PUBLISHER(Name, Address, Phone)

BOOK_COPIES(Book_id, Programme_id, No-of_Copies)

BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date)

LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address)

Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each programme, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2023 to Jun 2023.
3. Delete a book in the BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5. Create a view of all books and its number of copies that are currently available in the Library

2.2 Software Required:

Operating System: Windows 7

Software: Oracle 10g (SQL Plus)

2.3 Pre-Requisite:

Basics of DBMS, SQL Programming

2.4 Procedure:

TABLE CREATION

```
CREATE TABLE PUBLISHER (
NAME VARCHAR 2(20) PRIMARY KEY,
ADDRESS VARCHAR 2(20),
PHONE NUMBER (10));
```

```
CREATE TABLE BOOK (
BOOK_ID INT PRIMARY KEY,
TITLE VARCHAR 2(40),
PUBLISHER_NAME VARCHAR 2(20) REFERENCES PUBLISHER (NAME) ON DELETE CASCADE,
PUB_YEAR INT);
```

```
CREATE TABLE BOOK_AUTHORS (
BOOK_ID VARCHAR (20),
AUTHOR_NAME VARCHAR 2(20),
PRIMARY KEY (BOOK_ID, AUTHOR_NAME), FOREIGN KEY(BOOK_ID) REFERENCES BOOK ON DELETE CASCADE);
```

```
CREATE TABLE LIBRARY_PROGRAMME (
PROGRAMME_ID VARCHAR (20) PRIMARY KEY,
PROGRAMME_NAME VARCHAR (10),
ADDRESS VARCHAR (20));
```

```
CREATE TABLE BOOK_COPIES (
BOOK_ID VARCHAR (20) REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
 PROGRAMME_ID VARCHAR (20) REFERENCES LIBRARY_PROGRAMME (PROGRAMME_ID) ON DELETE CASCADE,
 NO_OF_COPIES NUMBER (2),
PRIMARY KEY (BOOK_ID, PROGRAMME_ID));
```

```
CREATE TABLE BOOK_LENDING (
BOOK_ID VARCHAR (20),
PROGRAMME_ID VARCHAR (20),
CARD_NO VARCHAR (20),
DATE_OUT DATE,
DUE_DATE DATE,
PRIMARY KEY (PROGRAMME_ID, BOOK_ID, CARD_NO),
FOREIGN KEY (BOOK_ID) REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY (PROGRAMME_ID) REFERENCES LIBRARY_PROGRAMM
(PROGRAMME_ID) ON DELETE CASCADE,
CONSTRAINT CK1 CHECK (DUE_DATE > DATE_OUT));

COMMIT;
Commit complete.
```

INSERTION OF VALUES:

```
SQL> INSERT INTO PUBLISHER VALUES ('&NAME','&ADDRESS','&PHONE');
Enter value for name: PEARSON

Enter value for address: LONDON Enter
value for phone: 9874522224

Old   1: INSERT INTO PUBLISHER VALUES ('&NAME','&ADDRESS','&PHONE')
New   1: INSERT INTO PUBLISHER VALUES ('PEARSON','LONDON',9874522224)

SQL> SELECT * FROM PUBLISHER;


        NAME                ADDRESS           PHONE
   ---------------     -------------------    ----------
       PEARSON              LONDON            9874522224
     TATAMCGRAW             NEWYORK           9858523565
       OXFORD                 UK              9885121112
      CAMBRIDGE               UK              9785634615
       OREILLY             CALIFORNIA         9994125455
```

```
SQL> INSERT INTO BOOK VALUES ('&BOOK_ID','&TITLE','&PUBLISHER_NAME',

'&PUB_YEAR');

Enter value for book_id: B101

Enter value for title: DBMS

Enter value for publisher_name: PEARSON

Enter value for pub_year: 2017

Old   1: INSERT INTO BOOK VALUES
('&BOOK_ID','&TITLE','&PUBLISHER_NAME','&PUB_YEAR')

New   1: INSERT INTO BOOK VALUES ('B101','DBMS','PEARSON',2017)
```

```
SQL> SELECT * FROM BOOK;

  BOOK_ID              TITLE            PUBLISHER_NAME        PUB_YEAR
--------------------  -----------------------  --------------------  ----------
    B101               DBMS               PEARSON               2017
    B102               AIML               TATAMCGRAW            2009
    B103               DCN                PEARSON               2017
    B104               ATC                OXFORD                2017
    B105               PYTHON             OREILLY               2014
    B106               HADOOP             PEARSON               2000


SQL> INSERT INTO BOOK_AUTHORS VALUES('& BOOK_ID','&AUTHOR_NAME');
Enter value for book_id: B101
Enter value for author_name: ELMARSI
Old   1: INSERT INTO BOOK_AUTHORS VALUES ('& BOOK_ID ','&AUTHOR_NAME')
New   1: INSERT INTO BOOK_AUTHORS VALUES ('B101', 'ELMARSI');
SQL> SELECT * FROM BOOK_AUTHORS;

              BOOK_ID               AUTHOR_NAME
        --------------------    --------------------
               B101                  ELMARSI
               B101                  NAVATHE
               B101                  RAMAKRISHNAN
               B102                  ELAINE
               B105                  SRINIVASAN
               B106                  DOUGLAS

SQL> INSERT INTO LIBRARY_PROGRAMME VALUES
('&PROGRAMME_ID','&PROGRAMME_NAME',’&ADDRESS');
 Enter value for programme_id: L1
 Enter value for programme_name: SAHYADRI
 Enter value for address: MANGALORE
```

```
Old   1: INSERT INTO LIBRARY_PROGRAMME VALUES
('&PROGRAMME_ID','&PROGRAMME_NAME','&ADDRESS')
New   1: INSERT INTO LIBRARY_PROGRAMME VALUES
('L1','SAHYADRI','MANGALORE');
SQL> SELECT * FROM LIBRARY_PROGRAMME;

          PROGRAMME_ID      PROGRAMME_NAME        ADDRESS
            -----------------      ---------------------------       -----------------

               L1                  SAHYADRI            MANGALORE

               L2                   SAPNA              MANGALORE

              L3                   SANKALP             BANGALORE

              L4                   PENGUIN              CHENNAI

             L5                     AGNES               CHENNAI


SQL> SET LINESIZE 200 PAGESIZE 3000;

SQL> INSERT INTO BOOK_COPIES VALUES
('&BOOK_ID','&PROGRAMME_ID','&NO_OF_COPIES');
 Enter value for book_id: B101
  Enter value for programme_id: L1

  Enter value for no_of_copies: 99

  Old   1: INSERT INTO BOOK_COPIES VALUES
('&BOOK_ID','&PROGRAMME_ID','&NO_OF_COPIES')
  New   1: INSERT INTO BOOK_COPIES VALUES('B101','L1',99)
```

```
SQL> SELECT * FROM BOOK_COPIES;

          BOOK_ ID   PROGRAMME_ID         NO_OF_COPIES

            -----          -----              ------------

            B101             L1                  99

            B102             L1                  99

            B103             L2                  99

            B103             L1                  99


SQL> INSERT INTO BOOK_LENDING VALUES
('&BOOK_ID','&PROGRAMME_ID','&CARD_NO','&DATE_OUT','&DUE_DATE');

Enter value for book_id: B101

Enter value for programme_id: L1

Enter value for card_no: FA101

Enter value for date_out: 02-JAN-21

Enter value for due_date: 09-JAN-21

Old   1: INSERT INTO BOOK_LENDING VALUES
('&BOOK_ID','&PROGRAMME_ID','&CARD_NO','&DATE_OUT','&DUE_DATE')

New   1: INSERT INTO BOOK_LENDING VALUES ('B101','L1','FA101','02-JAN-21','09-
JAN-21')
SQL> SELECT * FROM BOOK_LENDING;

  BOOK__ID    PROGRAMME__ID   CARD_NO    DATE_OUT     DUE_DATE
 -----             -----         ---------  --------      ---------
    B101             L1          FA101    02-JAN-21    09-JAN-21
    B101             L1          FA102    02-MAR-23    09-MAR-23
    B102             L1          FA102    02-MAR-23    09-MAR-23
    B101             L2          FA102    02-MAR-23    09-MAR-23
    B101             L1           S103    04-APR-22    30-JUN-22

SQL> COMMIT;

Commit complete.
```

2.5 RESULTS

QUERY 1:

```
SQL> SELECT B.BOOK_ID, TITLE, PUBLISHER_NAME, AUTHOR_NAME,
NO_OF_COPIES
FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES BC
WHERE B.BOOK_ID = BC.BOOK_ID AND
B.BOOK_ID = A.BOOK_ID;
```

```
BOOK_ID   BOOK_TITLE   PUBLISHER_NAME   AUTHOR_NAME    NO_OF_COPIES
-----     --------------------   ---------------   ---------------   ------------
B101      DBMS         PEARSON          ELMARSI        99
B101      DBMS         PEARSON          NAVATHE        99
B101      DBMS         PEARSON          RAMAKRISHNAN   99
B102      AIML         TATAMCGRAW       ELAINE         99

4 rows selected.
```

QUERY 2:

```
SQL> SELECT CARD_NO  FROM BOOK_LENDING
WHERE DATE_OUT
BETWEEN '01-JAN-2023' AND '30-JUN-2023'
GROUP BY CARD_NO
HAVING COUNT(*) >= 3 ;
```

```
CARD_NO
--------------------
FA102
```

QUERY 3:

```
SQL> SELECT * FROM BOOK;

 BOOK_ID         TITLE                                    PUBLISHER_NAME        PUB_YEAR
--------------------  ----------------------------------------      --------------------            ----------
   B101            DBMS                                       PEARSON                    2017
   B102            AIML                                       TATAMCGRAW                 2009
   B103            DCN                                        PEARSON                    2017
   B104            ATC                                        OXFORD                     2017
   B105           PYTHON                                      OREILLY                    2014
   B106           HADOOP                                       PEARSON                   2000

SQL> SELECT * FROM AUTHORS;

BOOK_ID             AUTHOR_NAME
--------------------      --------------------
B101                ELMARSI
B101                NAVATHE
B101                RAMAKRISHNAN
B102                ELAINE
B105                SRINIVASAN
B106                DOUGLAS

SQL> SELECT * FROM BOOK_COPIES;

BOOK_ ID     PROGRAMME_ID     NO_OF_COPIES
-----                  -----                 ------------

B101            L1                          99

B102            L1                          99

B103            L2                          99

B103            L1                          99
```

```
SQL> SELECT * FROM BOOK__LENDING;

  BOOK__ID   PROGRAMME__ID-    CARD_NO    DATE_OUT      DUE_DATE
                   ----        ---------
 -----                                    ---------     ---------
                    L1          FA101
    B101                        FA102     02-JAN-21     09-JAN-21
                    L1
    B101                                  02-MAR-23     09-MAR-23
    B102            L1          FA102     02-MAR-23     09-MAR-23
    B101            L2          FA102     02-MAR-23     09-MAR-23
    B101            L1          S103      04-APR-22     30-JUN-22
```

SQL> DELETE FROM BOOK WHERE BOOK_ID = &BID;

```
ENTER VALUE FOR BID: 'B103'

OLD   1: DELETE FROM BOOK WHERE BOOK_ID = &BID

NEW   1: DELETE FROM BOOK WHERE BOOK_ID = 'B103'


1 row deleted.

SQL> SELECT * FROM BOOK;
BOOK_ID          TITLE                            PUBLISHER_NAME     PUB_YEAR
---------------  -------------------------------  ------------------ ----------
B101               DBMS                           PEARSON              2017
B102              AIML                            TATAMCGRAW           2009
B104              ATC                             OXFORD               2017
B105              PYTHON                          OREILLY              2014
B106              HADOOP                          PEARSON              2000

SQL> SELECT * FROM BOOK_AUTHORS;

BOOK_ID             AUTHOR_NAME
------------------    ------------------
   B101                ELMARSI
   B101                NAVATHE
   B101                RAMAKRISHNAN
   B102                ELAINE
   B105                SRINIVASAN
   B106                DOUGLAS
```

```
SQL> SELECT * FROM BOOK_COPIES;

BOOK_ID            PROGRAMME_ID       NO_OF_COPIES
--------------------       --------------------            ------------
B101                          L1                 99
B102                          L1                 99


SQL> SELECT * FROM BOOK_LENDING;

BOOK_ID               PROGRAMME_ID        CARD_NO           DATE_OUT  DUE_DATE
--------------------        --------------------         --------------------       ---------          ---------
B101                    L1                   FA101             02-JAN-21     09-JAN-21
B101                    L1                   FA102             02-MAR-23    09-MAR-23
B102                    L1                  FA102              02-MAR-23    09-MAR-23
B101                    L2                  FA102              02-MAR-23    09-MAR-23
B101                    L1                  S103               04-APR-22     30-JUN-22
```

QUERY 4:

```
    CREATE TABLE BOOK1 (
    BOOK_ID VARCHAR2(20) PRIMARY KEY,
    TITLE VARCHAR2(40),
     PUBLISHER_NAME VARCHAR2(20) REFERENCES
     PUBLISHER(NAME) ON DELETE CASCADE,
     PUB_YEAR INT)
     PARTITION BY RANGE(PUB_YEAR)
     (PARTITION P1 VALUES LESS THAN(2001),
      PARTITION P2 VALUES LESS THAN(2005),
      PARTITION P3 VALUES LESS THAN (2010),
      PARTITION P4 VALUES LESS THAN(MAXVALUE));
    SQL> INSERT INTO BOOK1 VALUES
    ('&BOOK_ID','&TITLE','&PUBLISHER_NAME','&PUB_YEAR');
     Enter value for book_id: B101
     Enter value for title: DBMS
    Enter value for publisher_name: Pearson

    Enter value for pub_year: 2017

    Old   1: INSERT INTO BOOK1 VALUES

    ('&BOOK_ID','&TITLE','&PUBLISHER_NAME','&PUB_YEAR')

    NEW   1: INSERT INTO BOOK1 VALUES ('B101','DBMS','PEARSON',2017);


   SQL> SELECT * FROM BOOK1 PARTITION (P1);

BOOK_ID            TITLE                              PUBLISHER_NAME       PUB_YEAR
-------------------- ---------------------------------------- --------------------             ----------
B106               HADOOP                             PEARSON              2000


  SQL>SELECT * FROM BOOK1 PARTITION (P2);

BOOK_ID            TITLE                              PUBLISHER_NAME       PUB_YEAR
-------------------- ---------------------------------------- --------------------             ----------

no rows selected


   SQL> SELECT * FROM BOOK1 PARTITION (P3);

BOOK_ID            TITLE                                PUBLISHER_NAME       PUB_YEAR
-------------------- ----------------------------------------     --------------------             ----------

B102                   AIML                             TATAMCGRAW      2009



  SQL> SELECT * FROM BOOK1 PARTITION (P4);

BOOK_ID            TITLE                              PUBLISHER_NAME       PUB_YEAR

-------------------- ---------------------------------------- --------------------             ----------

B101          DBMS                     PEARSON             2017

B103          DCN                      PEARSON             2017

B104          ATC                      OXFORD              2017

B105          PYTHON                   OREILLY             2014
```

QUERY 5:

```
SQL> CREATE VIEW AVAILABLE_BOOK AS
SELECT B.BOOK_ID, B.TITLE, SUM (BC.NO_OF_COPIES)- (SELECT COUNT(*)
FROM BOOK_LENDING BL
WHERE BL.BOOK_ID= B. BOOK_ID
GROUP BY BL.BOOK_ID) AS BOOKS_AVAILABLE
FROM BOOK B, BOOK_COPIES BC
WHERE B.BOOK_ID=BC.BOOK_ID
GROUP BY B.BOOK_ID, B.TITLE;

 View created.
```

```
SQL>SELECT * FROM AVAILABLE_BOOK;

BOOK_ID          TITLE                              BOOKS_AVAILABLE
------------------- ---------------------------------------      ---------------
B101              DBMS                                  95
B102             AIML                                   98
```

2.6 Pre – Experiment Questions:

1. What is E-R model?
2. What is Object Oriented model?
3. What is an Entity?
4. What is an Entity type?
5. What is an Entity set?

2.7 Post – Experiment Questions:

1. What is an Extension of entity type?
2. What is an attribute?
3. What is a Relation Schema and a Relation?
4. What is degree of a Relation?
5. What is Relationship?

EXPERIMENT No. 03: College Database

3.1 Experiment Details:

Consider the schema for College database:

STUDENT(USN, SName, Address, Phone, Gender)

SEMSEC(SSID, Sem, Sec)

CLASS(USN, SSID)

COURSE(Subcode, Title, Sem, Credits)

IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to

1. List all the student details studying in the fourth semester "C" section
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN "4SF20CD001" in all courses.
4. Calculate the FinalIA (average of three test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:

   If FinalIA = 45 to 50 then CAT = "Outstanding"

   If FinalIA= 40 to 45 then CAT= "Good"

   If FinalIA = 30 to 40 then CAT = "Average"

   If FinalIA< 30 then CAT = "Weak"

   Give these details only for 8th semester A, B, and C section students.

3.2 Software Required:

Operating System: Windows 7

Software: Oracle 10g (SQL Plus)

3.3 Pre-Requisite:

Basics of DBMS, SQL Programming

3.4 Procedure:

TABLE CREATION:

```
CREATE TABLE STUDENT(
USN     VARCHAR2(20),
SNAME   VARCHAR2(10),
ADDRESS VARCHAR2(10),
PHONE   NUMBER(10),
GENDER  VARCHAR2(10),
PRIMARY KEY(USN));

CREATE TABLE SEMSEC(
SSID   NUMBER(5),
SEM   NUMBER(2),
SECTION  VARCHAR2(1),
PRIMARY KEY(SSID));

CREATE TABLE CLASS(
SSID NUMBER(5),
USN  VARCHAR2(20),
PRIMARY KEY(SSID,USN),
FOREIGN KEY(SSID) REFERENCES SEMSEC(SSID),
FOREIGN KEY(USN) REFERENCES STUDENT(USN));

CREATE TABLE COURSE(
SUBCODE VARCHAR2(7) PRIMARY KEY,
TITLE VARCHAR2(20),
SEM NUMBER(4),
CREDITS NUMBER(2));
```

```
CREATE TABLE IAMARKS(
USN  VARCHAR2(20),
SUBCODE VARCHAR2(7),
SSID NUMBER(5),
TEST1 NUMBER(3),
TEST2 NUMBER(3),
TEST3 NUMBER(3),
FINALIA NUMBER(3),
PRIMARY KEY(USN,SUBCODE,SSID),
FOREIGN KEY(USN) REFERENCES STUDENT(USN),
FOREIGN KEY(SUBCODE) REFERENCES COURSE(SUBCODE),
FOREIGN KEY(SSID) REFERENCES SEMSEC(SSID));

COMMIT;
```

Commit complete.

INSERTION OF VALUES:

```
SQL> INSERT INTO STUDENT VALUES
('&USN','&SNAME','&ADDRESS','&PHONE','&GENDER');

Enter value for usn: 4SF20CS089

Enter value for sname:  AJAY

Enter value for address: MANGALORE

Enter value for phone: 7338258875

Enter value for gender: MALE

old   1: INSERT INTO STUDENT

         VALUES('&USN','&SNAME','&ADDRESS','&PHONE','&GENDER')

new   1: INSERT INTO STUDENT VALUES('4SF20CS089',' AJAY

         ','MANGALORE','7338258875','MALE')


SQL> SELECT * FROM STUDENT;
USN                 SNAME       ADDRESS      PHONE        GENDER
-------------------  ----------  ----------   ----------   ----------
4SF20CS089     AJAY        MANGALORE    7338258875   MALE
4SF20IS109     VARSHINI    BANTWAL      8965239951   FEMALE
4SF20CS098     NAVISH      UDUPI        9956258987   MALE
4SF20CD001     SAHANA      SURATHKAL    8752683234   FEMALE
4SF20CS088     KAVYA       MANGALORE    7896523185   FEMALE


SQL> INSERT INTO SEMSEC VALUES('&SSID','&SEM','&SECTION');

Enter value for ssid: 1

Enter value for sem: 4

Enter value for section: A

old   1: INSERT INTO SEMSEC VALUES('&SSID','&SEM','&SECTION')

new   1: INSERT INTO SEMSEC VALUES('1','4 ','A')
```

```
SQL> SELECT * FROM SEMSEC;
      SSID        SEM         S
    ----------  ----------    -
         1          4         A
         2          4         C
         3          8         A
         4          8         B
         5          8         C
         6          5         A
```

```
SQL> INSERT INTO CLASS VALUES('&SSID','&USN');
Enter value for ssid: 1
Enter value for usn: 4SF20CD001
old   1: INSERT INTO CLASS VALUES('&SSID','&USN')
new   1: INSERT INTO CLASS VALUES('1','4SF20CD001')
```

```
SQL> SELECT * FROM CLASS;
      SSID     USN
  ----------   --------------------
       1       4SF20CD001
       2       4SF20CS088
       3       4SF20CS089
       4       4SF20IS109
       5       4SF20CS098
```

```
SQL> INSERT INTO COURSE VALUES('&SUBCODE','&TITLE','&SEM','&CREDITS');
Enter value for subcode: 20CS31
Enter value for title: DATA STRUCTURE
Enter value for sem: 4
Enter value for credits: 4
old   1: INSERT INTO COURSE VALUES('&SUBCODE','&TITLE','&SEM','&CREDITS')
new   1: INSERT INTO COURSE VALUES('20CS31 ','DATA STRUCTURE   ','4','4')
```

```
SQL> SELECT * FROM COURSE;

SUBCODE   TITLE                           SEM       CREDITS

-------   --------------------            ----------  ----------

20CS3       DATA STRUCTURE                  4           4

20CS32      UNIX                            4           3

20CS33      DBMS                            5           4

20CS34      DCN                             5           3

20CS35      AIML                            8           4
```

```
SQL>INSERT INTO IAMARKS
VALUES('&USN','&SUBCODE','&SSID','&TEST1','&TEST2','&TEST3','&FINALIA');
Enter value for usn: 4SF20CD001
Enter value for subcode: 20CS31
Enter value for ssid:1
Enter value for test1:38
Enter value for test2:35
Enter value for test3:32
Enter value for finalia:0
Old 1:INSERT INTO IAMARKS VALUES
      ('&USN','&SUBCODE','&SSID','&TEST1','&TEST2','&TEST3','&FINALIA');
New 1: INSERT INTO IAMARKS VALUES('4SF20CD001','20CS31 ',1,38,35,32,0);
```

```
SQL> SET LINESIZE 200 PAGESIZE 3000;

SQL> SELECT * FROM IAMARKS;

USN                   SUBCODE     SSID          TEST1    TEST2        TEST       FINALIA

--------------------  -------     ----------    ----------  ----------   ---------   ----------

4SF20CD001            20CS32        1             28        26           29          0

4SF20CS088            20CS31        2             38        42           32          0

4SF20CS089            20CS33        3             42        46           41          0

4SF20CS098            20CS34        4             48        46           50          0

4SF20IS109            20CS35        5             28        26           29          0
```

3.5 RESULTS

QUERY 1:

```
SQL>  SELECT A.*,B.SEM,B.SECTION FROM STUDENT A,SEMSEC B,CLASS C
WHERE      A.USN=C.USN AND B.SSID=C.SSID  AND B.SEM=4 AND B.SECTION='C';

USN                 SNAME     ADDRESS         PHONE        GENDER         SEM      S
-------------------- ----------       ----------       ----------        ----------       ----------     -
4SF20CS088       KAVYA    MANGALORE   7896523185     FEMALE           4         C
```

QUERY 2:

```
SQL> SELECT SEM,SECTION,GENDER,COUNT(*) FROM STUDENT S, SEMSEC
S1,CLASS C  WHERE S.USN=C.USN AND  S1.SSID=C.SSID GROUP BY
(GENDER,SEM,SECTION) ORDER BY(SEM);


SEM      S     GENDER      COUNT(*)
---------- -     ----------        ----------
 4        A    FEMALE           1
 4        C    FEMALE           1
 8        B    FEMALE           1
 8        A     MALE             1
 8        C     MALE             1
```

QUERY 3:

```
CREATE VIEW TEST_MARKS
AS SELECT SUBCODE ,TEST1 FROM IAMARKS
WHERE USN='4SF20CD001';


View created.
SQL> SELECT  * FROM TEST_MARKS;


SUBCODE     TEST1

-------              ----------

20CS32              28
```

QUERY 4:

```
SQL>  UPDATE IAMARKS SET
FINALIA=(TEST1+TEST2+ TEST3)/3;

5 rows updated.
SQL> SELECT * FROM IAMARKS;
USN                 SUBCODE      SSID     TEST1      TEST2      TEST3    FINALIA
--------------------    -------         ---------- ----------      ----------    ----------    ----------
4SF20CD001       20CS32           1           28            26          29           29
4SF20CS088       20CS31           2           38            42          32           40
4SF20CS089       20CS33           3           42            46          41           44
4SF20CS098       20CS34           4           48            46          50           49
4SF20IS109       20CS35            5          28            26          29            29
```

QUERY 5:

```
SQL> SELECT USN,FINALIA,
CASE
WHEN FINALIA BETWEEN 45 AND 50 THEN 'OUTSTANDING'
WHEN FINALIA BETWEEN 40 AND 45 THEN 'GOOD'
WHEN FINALIA BETWEEN 30 AND 40 THEN 'AVERAGE'
WHEN FINALIA <30  THEN 'WEAK'
END
AS CATEGORY FROM IAMARKS I,
SEMSEC S WHERE I.SSID=S.SSID
AND SEM=8 AND SECTION IN('A','B','C');

USN                  FINALIA  CATEGORY
--------------------   ----------    -----------
4SF20CS089            44          GOOD
4SF20CS098            49          OUTSTANDING
4SF20IS109           29 WEAK
```

3.6 Pre – Experiment Questions:

1. When is a functional dependency F said to be minimal?
2. What is multivalued dependency?
3. What is 1 NF (Normal Form), 2NF, 3NF?
4. What is Lossless join property?
5. What is Fully Functional dependency?

3.7 Post – Experiment Questions:

1. What is BCNF (Boyce-Codd Normal Form)?
2. What is 4NF?
3. What is 5NF?
4. What is Domain-Key Normal Form?
5. What are partial, alternate, artificial, compound and natural key?

EXPERIMENT No. 04: Company Database

4.1 Experiment Details:

Consider the schema for Company database:

EMPLOYEE (Eid, Name, Address, Gender, Salary, SuperEid, Dno)

DEPARTMENT (Dnum, Dname, DMgr_id, Mgr_start_date)

DLOCATION (Dno, Dlocation)

PROJECT (Pnum, Pname, Plocation, Dno)

WORKS_ON (Eid, Pno, Hours)

DEPENDENT (Empid, Dep_name, Gender, Bdate, Relationship)

Write SQL queries to

1. Make a list of all project numbers for projects that involve an employee whose name is “Rahul”, either as a worker or as a manager of the department that controls the project.
2. Show the resulting salaries if every employee working on the “IoT” project is given a 10 percent raise.
3. Find the sum of the salaries of all employees of the “Accounts” department, as well as the maximum salary, the minimum salary, and the average salary in this department.
4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).
5. Create a view Dept_info that gives details of department name, number of employees and total salary of each department.

4.2 Software Required:

Operating System: Windows 7

Software: Oracle 10g (SQL Plus)

4.3 Pre-Requisite:

Basics of DBMS, SQL Programming

4.4 Procedure:

TABLE CREATION:

```
CREATE TABLE EMPLOYEE(
EID INT PRIMARY KEY,
NAME VARCHAR2(20),
ADDRESS VARCHAR2(20),
GENDER CHAR(1) CHECK(GENDER ='M' OR GENDER ='F'),
SALARY NUMBER(6),
SUPEREID REFERENCES EMPLOYEE(EID),
DNO NUMBER);

CREATE TABLE DEPARTMENT (
DNUM NUMBER(5) PRIMARY KEY,
DNAME VARCHAR2(10) ,
DMGR_ID REFERENCES EMPLOYEE(EID),
MGR_START_DATE DATE);

CREATE TABLE DLOCATION (
DNO REFERENCES DEPARTMENT(DNUM),
DLOCATION  VARCHAR2(10),
PRIMARY KEY(DNO,LOCATION));

CREATE TABLE PROJECT(
PNUM NUMBER(2) PRIMARY KEY,
PNAME VARCHAR2(20),
PLOCATION VARCHAR2(20),
DNO NUMBER  REFERENCES    DEPARTMENT(DNUM) );

CREATE TABLE  WORKS_ON (
EID  NUMBER(5) REFERENCES EMPLOYEE(EID),
PNO  NUMBER(2) REFERENCES   PROJECT(PNUM),
HOURS NUMBER(5,2),
PRIMARY KEY(EID,PNO));
```

```
CREATE TABLE DEPENDENT(
EMPID INT CONSTRAINT DEP_EMPID_PK PRIMARY KEY,
DEP_NAME VARCHAR2(12),
GENDER VARCHAR2(5),
BDATE DATE,
RELATIONSHIP VARCHAR2(12),
FOREIGN KEY(EMPID)REFERENCES EMPLOYEE(EID) ON DELETE CASCADE);
```

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT FK FOREIGN KEY(DNO) REFERENCES
DEPARTMENT(DNUM);
```

```
Table altered.
```

```
COMMIT;
```

Commit complete.

INSERTION OF VALUES:

```
SQL> INSERT INTO EMPLOYEE VALUES
('&EID','&NAME','&ADDRESS','&GENDER','&SALARY','&SUPEREID',
'&DNO');
Enter value for eid: 1
Enter value for name: RAHUL
Enter value for address: MANGALURU
Enter value for gender: M
Enter value for salary: 35000
Enter value for supereid:1
Enter value for dno: NULL
Old   1:  INSERT INTO EMPLOYEE VALUES
         ('&EID','&NAME','&ADDRESS','&GENDER','&SALARY','&SUPEREID','&DNO')
New   1: INSERT INTO EMPLOYEE VALUES(1, 'RAHUL' , 'MANGALURU',
         'M',35000,1,NULL);

SQL> SELECT * FROM EMPLOYEE;

EID NAME            ADDRESS             G            SALARY   SUPEREID      DNO
----------          -------------------- -------------------- -  ----------     ----------     ----------
 1 RAHUL             MANGALURU          M            35000          1
 2 SAHANA           MANGALURU          F            35000          1
 3 SAGAR             BENGALURU          M           35000          1
 4 SAGARIK          MANGALURU          M            35000          1
 5 SAJAAN           MYSORE              M            600000         1

SQL> INSERT INTO DEPARTMENT VALUES
('&DNUM','&DNAME','&DMGR_ID','&MGR_START_DATE');
Enter value for dnum: 1
Enter value for dname: CSE
Enter value for dmgr_id:1
Enter value for mgr_start_date: 2-NOV-2007
```

```
Old   1: INSERT INTO DEPARTMENT VALUES
('&DNUM','&DNAME','&DMGR_ID','&MGR_START_DATE')
New   1: INSERT INTO DEPARTMENT VALUES(1,'CSE',1,'2-NOV-2007');

SQL> SELECT * FROM DEPARTMENT;

     DNUM      DNAME        DMGR_ID        MGR_START
     ----------   ----------      ----------         ---------
        1          CSE              1            02-NOV-07
        2           IOT             2            02-NOV-07
        3           ACCOUNT         2            02-NOV-17
        4           ISE             1            02-NOV-00
        5           FINANCE        1             03-NOV-01

ALTER TABLE EMPLOYEE ADD CONSTRAINT FK FOREIGN KEY(DNO) REFERENCES
DEPARTMENT(DNUM);

Table altered.

SQL> UPDATE EMPLOYEE SET DNO=4 WHERE EID=1;
SQL>UPDATE EMPLOYEE SET DNO=1WHERE EID=2;
SQL>UPDATE EMPLOYEE SET DNO=3 WHERE EID=3;
SQL>UPDATE EMPLOYEE SET DNO=3 WHERE EID=4;
SQL>UPDATE EMPLOYEE SET DNO=3 WHERE EID=5;


SQL> SELECT * FROM EMPLOYEE;

EID      NAME             ADDRESS          GENDER    SALARY   SUPERID    DNO

   1    RAHUL          MANGALURU          M         35000         1          4
   2    SAHANA         MANGALURU          F         35000         1          1
   3    SAGAR          BENGALURU          M         35000         1          3
   4    SAGARIK        MANGALURU          M         35000         1          3
   5    SAJAAN         MYSORE             M        600000         1          3
```

```
SQL> SELECT * FROM DEPARTMENT;

      DNUM              DNAME                MGR_SSN     MGR_START

      -------         ------------------------     ----------         ---------
        1                CSE                  1          02-NOV-07
        2                IOT                  2          02-NOV-07
        3              ACCOUNT                2          02-NOV-17
        4                ISE                  1          02-NOV-00
        5              FINANCE                1          03-NOV-01

SQL> INSERT INTO DLOCATION VALUES ('&DNO','&DLOCATION');
Enter value for dno: 1
Enter value for dlocation: MANGALURU

Old   1: INSERT INTO DLOCATION VALUES ('&DNO','&DLOCATION')
New   1: INSERT INTO DLOCATION VALUES (1,'MANGALURU')

SQL> SELECT * FROM DLOCATION;
     DNO              DLOCATION
     ----------           --------------------
        1             MANGALURU
        1            MYSORE
        2            MANGALURU
        3            BENGALURU
        4            MANGALURU
        5            MANGALURU

SQL> INSERT INTO PROJECT VALUES ('&PNUM','&PNAME','&PLOACTION','&DNO);
Enter value for pnum: 2
Enter value for pname: DATA MINING
Enter value for ploaction: 'MANAGLURU
Enter value for dno: 1
Old   1: INSERT INTO PROJECT VALUES ('&PNUM','&PNAME','&PLOACTION','&DNO')
New   1: INSERT INTO PROJECT VALUES (2,'DATA MINING','MANAGLURU',1);
```

```
SQL> SELECT * FROM PROJECT;

    PNUM  PNAME                           PLOCATION          DNO
    ------ --------------------------------  ------------------  -------
       2  DATA MINING                        MANAGLURU          1
       1  IOT                                MANAGLURU          1
       3  CC                                   HUBLI            3
       4  IMAGE PROCESSING                   MANAGLURU          4
       5  RESEARCH                           MANAGLURU          5

SQL> INSERT INTO WORKS_ON VALUES ('&EID','&PNO','&HOURS');
Enter value for eid: 1
Enter value for pno: 1
Enter value for hours: 4
Old   1: INSERT INTO WORKS_ON VALUES ('&EID','&PNO','&HOURS')
New   1: INSERT INTO WORKS_ON VALUES (1,1,4);

SQL> SELECT * FROM WORKS_ON;

EID       PNO     HOURS
------    -------  ----------
   1        1        4
   2        1        5
   3        2        4
   4        3        4
   5        5        4

SQL> INSERT INTO DEPENDENT VALUES ('& EMPID','& DEP_NAME','& GENDER', '&
BDATE', '& RELATIONSHIP');
Enter value for empid: 1
Enter value for dep_name : SHREYA
Enter value for gender: F
Enter value for bdate: 22-JAN-1975
Enter value for relationship: WIFE
```

```
Old   1: INSERT INTO DEPENDENTVALUES ('& EMPID','& DEP_NAME','& GENDER', '&
BDATE', '& RELATIONSHIP')
New   1: INSERT INTO DEPENDENT  VALUES(1,'SHREYA','F','22-JAN-1975','WIFE')

SQL> SELECT * FROM DEPENDENT;

    EMPID    DEP_NAME     GENDER     BDATE     RELATIONSHIP
    ----------     ------------          -----          ---------        ------------
       1          SHREYA               F        22-JAN-75     WIFE
       2          AKASH                M        15-JUN-77      BROTHER
       3          PRIYA                F        10-SEP-80     SISTER
       4          NIKIL                M        02-FEB-79      FATHER
       5          AKSHA                F        30-OCT-75     MOTHER

COMMIT;
```

Commit complete.

4.5 RESULTS:

QUERY 1:

```
SELECT PNO
FROM WORKS_ON
WHERE EID IN
(SELECT EID FROM EMPLOYEE WHERE NAME='RAHUL')
UNION
SELECT PNUM
FROM PROJECT
WHERE DNO IN
(SELECT DNUM FROM DEPARTMENT WHERE DMGR_ID IN
(SELECT EID FROM EMPLOYEE WHERE NAME='RAHUL'));
```

```
                  PNO
             ----------
                   1
                   2
                   4
                   5
```

QUERY 2:

```
SELECT EID,NAME, SALARY,SALARY+0.1*SALARY AS UPDATED_SALARY
FROM EMPLOYEE
WHERE EID IN
(SELECT EID FROM WORKS_ON WHERE PNO IN(
SELECT PNUM FROM PROJECT WHERE PNAME='IOT'));
```

```
     EID          NAME              SALARY     UPDATED_SALARY
  ----------   --------------------   ----------    --------------
      1           RAHUL                35000        38500
      2           SAHANA                35000        38500
```

QUERY 3:

```
SELECT SUM(SALARY), AVG(SALARY), MAX(SALARY), MIN(SALARY)
FROM EMPLOYEE E,DEPARTMENT D
WHERE D.DNUM=E.DNO AND DNAME='ACCOUNT';
```

```
SUM(SALARY)     AVG(SALARY)     MAX(SALARY)     MIN(SALARY)
-----------     -----------     -----------     -----------
 670000         223333.333      600000          35000
```

QUERY 4:

```
SELECT EID,NAME
FROM EMPLOYEE E
WHERE NOT EXISTS(
(SELECT PNUM FROM PROJECT WHERE DNO=5) MINUS
(SELECT PNO FROM WORKS_ON W WHERE W.EID=E.EID) );
```

```
   EID      NAME
----------  --------------------
    5       SAJAAN
```

QUERY 5:

```
CREATE VIEW DEPT_INFO(NAME,COUNT_EMP,SUM_SAL) AS
SELECT D.DNAME, COUNT(*), SUM(SALARY)
FROM DEPARTMENT D INNER JOIN EMPLOYEE E
ON E.DNO = D.DNUM
GROUP BY D.DNAME;
```

```
NAME       COUNT_EMP    SUM_SAL
----------    ----------   ----------
ISE           1             35000
ACCOUNT       3             670000
CSE           1             35000
```

4.6 Pre – Experiment Questions:

1. What is indexing and what are the different kinds of indexing?
2. What is system catalog or catalog relation? How is better known as?
3. What is meant by query optimization?
4. What is join dependency and inclusion dependency?

4.7 Post – Experiment Questions:

1. What is durability in DBMS?
2. What do you mean by atomicity and aggregation?
3. What is a checkpoint and when does it occur?
4. What are the different phases of transaction?
5. What is "transparent DBMS"?

EXPERIMENT No. 05: Airline Database

5.1 Experiment Details:

Consider the schema for Airline database:

Flights (Flight_num, Source, Destination, Distance, Departs, Arrives, Price)

Aircraft (Aid, Aname, Cruising_range)

Certified (Emp_id, Aid)

Employees (Emp_id, Ename, Salary)

Note: The Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write SQL queries to

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
2. For each pilot who is certified for more than three aircrafts, find the Emp_id and the maximum cruising range of the aircraft for which she or he is certified.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Mumbai.
4. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
5. Find the employee name and salary earning the second highest salary.

5.2 Software Required:

Operating System: Windows 7

Software: Oracle 10g (SQL Plus)

5.3 Pre-Requisite:

Basics of DBMS, SQL Programming

5.4 Procedure:

TABLE CREATION:

```
CREATE TABLE FLIGHT(
FLIGHT_NUM INT,
SOURCE VARCHAR(20),
DESTINATION VARCHAR(20),
DISTANCE INT,
DEPARTS VARCHAR(10),
ARRIVES VARCHAR(10),
PRICE INT,
PRIMARY KEY (FLIGHT_NUM) );

CREATE TABLE EMPLOYEES(
EMP_ID INT,
ENAME VARCHAR(20),
SALARY INT,
PRIMARY KEY (EMP_ID) );

CREATE TABLE AIRCRAFT(
AID INT,
ANAME VARCHAR(20),
CRUISING_RANGE INT,
PRIMARY KEY (AID) );

CREATE TABLE CERTIFIED(
EMP_ID INT,
AID INT,
PRIMARY KEY (EMP_ID,AID),
FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEES (EMP_ID),
FOREIGN KEY (AID) REFERENCES AIRCRAFT (AID) );
```

INSERTION OF VALUES:

```
SQL> INSERT INTO FLIGHT VALUES('&FLIGHT_NUM','&SOURCE','&DESTINATION
','&DISTANCE','&DEPARTS','&ARRIVES','&PRICE');
Enter value for flight_num: 1
Enter value for source: BANGALORE
Enter value for destination: MANGALORE
Enter value for distance: 300
Enter value for departs: 10:45
Enter value for arrives: 12:00
Enter value for price: 10000
old   1: INSERT INTO FLIGHT VALUES('&FLIGHT_NUM','&SOURCE','&DESTINATION
','&DISTANCE','&DEPARTS','& ARRIVES','&PRICE')
new   1: INSERT INTO FLIGHT VALUES('1','BANGALORE','MANGALORE
','300','10:45','12:00','10000')
SQL> SET LINESIZE 200 PAGESIZE 3000;
SQL> SELECT * FROM FLIGHT;
FLIGHT_NUM    SOURCE        DESTINATION     DISTANCE   DEPARTS    ARRIVES      PRICE
----------    -------------------- --------------------   ----------   ----------   ----------   ----------
      1      BANGALORE     MANGALORE          300        10:45      12:00      10000
      2      BANGALORE     DELHI             5000        12:15       4:30      25000
      3      BANGALORE     MUMBAI            3500         2:15       5:25      30000
      4      DELHI         MUMBAI            4500        10:15      12:05      35000
      5      DELHI         FRANKFURT        18000         7:15       5:30      90000
      6      BANGALORE     FRANKFURT        19500        10:00       7:45      95000
      7      BANGALORE     FRANKFURT        17000        12:00       6:30      99000
```

```
SQL> INSERT INTO AIRCRAFT VALUES('&AID','&ANAME','&CRUISING_RANGE');
Enter value for aid: 123
Enter value for aname: AIRBUS
Enter value for cruising_range: 1000
old   1: INSERT INTO AIRCRAFT VALUES('&AID','&ANAME','&CRUISING_RANGE')
new   1: INSERT INTO AIRCRAFT VALUES('123','AIRBUS','1000')
```

```
SQL> SELECT * FROM AIRCRAFT;

 AID          ANAME          CRUISING_RANGE
----------   --------------------   -------------
 123         AIRBUS                  1000
 302         BOEING                  5000
 306         JET01                   5000
 378         AIRBUS380               8000
 456         AIRCRAFT                 500
 789         AIRCRAFT02               800
 951         AIRCRAFT03              1000

SQL> INSERT INTO EMPLOYEES VALUES('&EMP_ID','&ENAME','&SALARY');
Enter value for emp_id: 1
Enter value for ename: AJAY
Enter value for salary: 30000
old   1: INSERT INTO EMPLOYEES VALUES('&EMP_ID','&ENAME','&SALARY')
new   1: INSERT INTO EMPLOYEES VALUES('1','AJAY','30000')

SQL> SELECT * FROM EMPLOYEES;

 EMP_ID        ENAME          SALARY
  ----------   --------------------   ----------
    1        AJAY             30000
    2        AJITH            85000
    3        ARNAB            50000
    4        HARRY            45000
    5        ARUN             90000
    6        JOSH             75000
    7        RAM             100000
```

```
SQL> INSERT INTO CERTIFIED VALUES('&EMP_ID','&AID');
Enter value for emp_id: 1
Enter value for aid: 123
old   1: INSERT INTO CERTIFIED VALUES('&EMP_ID','&AID')
new   1: INSERT INTO CERTIFIED VALUES('1','123')
SQL> SELECT * FROM CERTIFIED;
         EMP_ID        AID
         ----------    ----------
           1             123
           1             302
           1             306
           1             378
           1            789
           1            951
           2           123
           2           306
           2           378
           3          456
           3          951
           4          378
           5          302
           5          789
           6         456
           6         789
           7         302
```

5.5 RESULTS:

QUERY 1:

```
SQL> SELECT DISTINCT A.ANAME
FROM AIRCRAFT A, CERTIFIED C, EMPLOYEES E
WHERE A.AID=C.AID AND C.EMP_ID=E.EMP_ID
AND NOT EXISTS
(SELECT *
    FROM EMPLOYEES E1
    WHERE E1.EMP_ID=E.EMP_ID
    AND E1.SALARY<80000);

          ANAME
          --------------------
          AIRBUS
          AIRCRAFT02
          BOEING
          JET01
          AIRBUS380
```

QUERY 2:

```
SQL> SELECT C.EMP_ID,MAX(CRUISING_RANGE)
FROM CERTIFIED C,AIRCRAFT A
WHERE C.AID=A.AID
GROUP BY C.EMP_ID
HAVING COUNT(*)>3;

    EMP_ID     MAX(CRUISING_RANGE)
   ----------  ------------------
         1            8000
```

QUERY 3:

```
SQL> SELECT DISTINCT E.ENAME
FROM EMPLOYEES E
WHERE E.SALARY<=
(SELECT MIN(F.PRICE)
    FROM FLIGHT F
    WHERE F.SOURCE='BANGALORE'
    AND F.DESTINATION='MUMBAI');

            ENAME
            --------------------
            AJAY
```

QUERY 4:

```
SQL> SELECT A.AID
FROM AIRCRAFT A
WHERE A.CRUISINGRANGE>
(SELECT MIN(F.DISTANCE)
    FROM FLIGHT F
    WHERE F.SOURCE='BANGALORE'
    AND F.DESTINATION='DELHI');
                    AID
                    ----------
                    378
```

QUERY 5:

```
SQL> SELECT ENAME,SALARY FROM EMPLOYEES WHERE SALARY=(SELECT
        MAX(SALARY)
        FROM EMPLOYEES
        WHERE SALARY <(SELECT MAX(SALARY)FROM EMPLOYEES));

      ENAME                 SALARY
       --------------------   ----------
       ARUN                  90000
```

5.6 Pre – Experiment Questions:

1. What is database?
2. What is DBMS?
3. What is a Database system?
4. Advantages of DBMS?

5.7 Post – Experiment Questions:

1. Define the "integrity rules"
2. What is extension and intension?
3. What is Data Independence?
4. What is Data Model?