**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW): Lab on DBMS**

# Duration 3:00 Hrs. Max marks: 20

# Instructions:

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

SUPPLIER ( Sno, Sname, address,City)

**CREATE TABLE SUPPLIER (**

**Sno INT PRIMARY KEY,**

**Sname VARCHAR(100) NOT NULL,**

**address VARCHAR(100) NOT NULL,**

**City VARCHAR(100) NOT NULL**

**);**

**INSERT INTO SUPPLIER (Sno, Sname, address, City) VALUES**

**(1, 'Supplier 1', 'Address 1', 'London'),**

**(2, 'Supplier 2', 'Address 2', 'Paris'),**

**(3, 'Supplier 3', 'Address 3', 'Rome'),**

**(4, 'Supplier 4', 'Address 4', 'New York'),**

**(5, 'Supplier 5', 'Address 5', 'Amsterdam');**

PARTS ( Pno, Pname, Color, Weight, price )

**CREATE TABLE PARTS (**

**Pno INT PRIMARY KEY,**

**Pname VARCHAR(100) NOT NULL,**

**Color VARCHAR(50) NOT NULL,**

**Weight DECIMAL(10, 2) NOT NULL,**

**price DECIMAL(10, 2) NOT NULL**

**);**

**INSERT INTO PARTS (Pno, Pname, Color, Weight, price) VALUES**

**(1, 'Part 1', 'Red', 10.5, 20.00),**

**(2, 'Part 2', 'Blue', 15.2, 30.00),**

**(3, 'Part 3', 'Green', 12.8, 25.50),**

**(4, 'Part 4', 'Yellow', 8.5, 18.75),**

**(5, 'Part 5', 'Black', 11.0, 35.00);**

PROJECT ( Jno, Mame, City )

**CREATE TABLE PROJECT (**

**Jno INT PRIMARY KEY,**

**Jname VARCHAR(100) NOT NULL,**

**City VARCHAR(100) NOT NULL,**

**CONSTRAINT CHK\_City CHECK (City IN ('London', 'Paris', 'Rome', 'New York', 'Amsterdam'))**

**);**

**INSERT INTO PROJECT (Jno, Jname, City) VALUES**

**(1, 'Project 1', 'London'),**

**(2, 'Project 2', 'Paris'),**

**(3, 'Project 3', 'Rome'),**

**(4, 'Project 4', 'New York'),**

**(5, 'Project 5', 'Amsterdam');**

SPJ ( in=q,En, JUQ, Qty )

**CREATE TABLE SPJ (**

**Sno INT,**

**Pno INT,**

**Jno INT,**

**Qty INT NOT NULL,**

**CONSTRAINT FK\_SUPPLIER FOREIGN KEY (Sno) REFERENCES SUPPLIER(Sno),**

**CONSTRAINT FK\_PARTS FOREIGN KEY (Pno) REFERENCES PARTS(Pno),**

**CONSTRAINT FK\_PROJECT FOREIGN KEY (Jno) REFERENCES PROJECT(Jno),**

**CONSTRAINT CHK\_Qty CHECK (Qty >= 0)**

**);**

**INSERT INTO SPJ (Sno, Pno, Jno, Qty) VALUES**

**(1, 1, 1, 5),**

**(2, 2, 1, 3),**

**(3, 3, 1, 4),**

**(4, 4, 2, 2),**

**(5, 5, 2, 7),**

**(1, 1, 2, 3),**

**(2, 2, 2, 6),**

**(3, 3, 2, 1),**

**(4, 4, 3, 4),**

**(5, 5, 3, 2);**

**Integrity Constraints:**

* The values of any attributes should not be null.
* Legal cities are London, Paris, Rome, New York and Amsterdam.
* Supplier Number must start with 'S' followed by a decimal integer in the range of 0 to 9999.

**Queries:**

1. Find all the projects which are provided 3 or more parts .

**SELECT Jname**

**FROM PROJECT**

**WHERE Jno IN (**

**SELECT Jno**

**FROM SPJ**

**GROUP BY Jno**

**HAVING COUNT(\*) >= 3**

**);**

1. Write a trigger on PROJECT table for update / insert such that the .jname value Should not be repeated.

**CREATE TRIGGER trg\_Project\_Unique\_Jname**

**BEFORE INSERT OR UPDATE ON PROJECT**

**FOR EACH ROW**

**BEGIN**

**DECLARE cnt INT;**

**SET cnt = (**

**SELECT COUNT(\*)**

**FROM PROJECT**

**WHERE Jname = NEW.Jname**

**);**

**IF cnt > 0 THEN**

**SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Jname value already exists.';**

**END IF;**

**END;**

1. Find full details of all projects in London.

**SELECT \***

**FROM PROJECT**

**WHERE City = 'London'**;

1. Write a procedure for calculating the total sales of all the parts which are provided to projects in paris city.

**DELIMITER //**

**CREATE PROCEDURE CalculateTotalSales(INOUT total DECIMAL(10, 2))**

**BEGIN**

**SELECT SUM(SPJ.Qty \* PARTS.price)**

**INTO total**

**FROM SPJ**

**INNER JOIN PARTS ON SPJ.Pno = PARTS.Pno**

**INNER JOIN PROJECT ON SPJ.Jno = PROJECT.Jno**

**WHERE PROJECT.City = 'Paris';**

**END//**

**DELIMITER ;**

**-- Usage:**

**SET @totalSales = 0;**

**CALL CalculateTotalSales(@totalSales);**

**SELECT @totalSales;**

Design an Input form for entering Parts data. Apply possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks: 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

PRODUCT ( Maker, Modelno, Type )

PC ( Modelno,,Speed, RAM, HD, CD, Price )

LAPTOP ( Modelno, Speed, RAM, HD, Price ) PRINTER ( Mg\_delaQ, Color, Type, Price )

**Details regarding Schemas**

* PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.
* The value for Maker in Product table can be IBM, Compaq,etc.
* PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black 8r. white), type(laser, ink-jet, dot- matrix or dry), and price.

**Integrity Constraints:**

* The values of any attributes should not be null.
* Product Type should one of these (PC, Laptop or Printer)

-- Create tables

CREATE TABLE PRODUCT (

Maker VARCHAR(100) NOT NULL,

Modelno INT PRIMARY KEY,

Type VARCHAR(50) NOT NULL,

CONSTRAINT CHK\_Type CHECK (Type IN ('PC', 'Laptop', 'Printer'))

);

CREATE TABLE PC (

Modelno INT PRIMARY KEY,

Speed INT NOT NULL,

RAM INT NOT NULL,

HD INT NOT NULL,

CD INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

CONSTRAINT FK\_PC\_Product FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)

);

CREATE TABLE LAPTOP (

Modelno INT PRIMARY KEY,

Speed INT NOT NULL,

RAM INT NOT NULL,

HD INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

CONSTRAINT FK\_Laptop\_Product FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)

);

CREATE TABLE PRINTER (

Modelno INT PRIMARY KEY,

Color CHAR(1) NOT NULL,

Type VARCHAR(50) NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_Color CHECK (Color IN ('T', 'F')),

CONSTRAINT FK\_Printer\_Product FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)

);

-- Insert data into tables

INSERT INTO PRODUCT (Maker, Modelno, Type) VALUES

('IBM', 1, 'PC'),

('Compaq', 2, 'PC'),

('Dell', 3, 'Laptop'),

('HP', 4, 'Laptop'),

('Epson', 5, 'Printer');

INSERT INTO PC (Modelno, Speed, RAM, HD, CD, Price) VALUES

(1, 300, 4, 500, 52, 699.99),

(2, 250, 8, 1000, 40, 999.99),

(3, 200, 16, 2000, 36, 1499.99);

INSERT INTO LAPTOP (Modelno, Speed, RAM, HD, Price) VALUES

(3, 1800, 8, 256, 1299.99),

(4, 2200, 16, 512, 1999.99);

INSERT INTO PRINTER (Modelno, Color, Type, Price) VALUES

(5, 'T', 'Laser', 399.99),

(6, 'F', 'Dot-Matrix', 199.99),

(7, 'T', 'Ink-Jet', 299.99);

-- Integrity Constraints

ALTER TABLE PC ADD CONSTRAINT CHK\_PC\_Speed CHECK (Speed >= 150);

ALTER TABLE PC ADD CONSTRAINT CHK\_PC\_RAM CHECK (RAM > 0);

ALTER TABLE PC ADD CONSTRAINT CHK\_PC\_HD CHECK (HD > 0);

ALTER TABLE PC ADD CONSTRAINT CHK\_PC\_CD CHECK (CD > 0);

ALTER TABLE PC ADD CONSTRAINT CHK\_PC\_Price CHECK (Price >= 0);

ALTER TABLE LAPTOP ADD CONSTRAINT CHK\_Laptop\_Speed CHECK (Speed > 0);

ALTER TABLE LAPTOP ADD CONSTRAINT CHK\_Laptop\_RAM CHECK (RAM > 0);

ALTER TABLE LAPTOP ADD CONSTRAINT CHK\_Laptop\_HD CHECK (HD > 0);

ALTER TABLE LAPTOP ADD CONSTRAINT CHK\_Laptop\_Price CHECK (Price >= 30000);

ALTER TABLE PRINTER ADD CONSTRAINT CHK\_Printer\_Price CHECK (Price >= 0);

**Queries:**

* 1. Find PC models having a speed of at least 150 MHz.

SELECT \*

FROM PC

WHERE Speed >= 150;

* 1. Find those manufacturers that sell Laptops, but not PC's.

SELECT DISTINCT Maker

FROM PRODUCT

WHERE Type = 'Laptop' AND Maker NOT IN (

SELECT DISTINCT Maker

FROM PRODUCT

WHERE Type = 'PC'

);

* 1. Write a trigger on LAPTOP table such that the price should not less than

30000

CREATE TRIGGER trg\_Laptop\_Price

BEFORE INSERT OR UPDATE ON LAPTOP

FOR EACH ROW

BEGIN

IF NEW.Price < 30000 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Laptop price should not be less than 30000.';

END IF;

END;

* 1. Write a procedure to find the manufacturer who has produced the most expensive laptop.

DELIMITER //

CREATE PROCEDURE FindMostExpensiveLaptopManufacturer(OUT manufacturer VARCHAR(100))

BEGIN

SELECT Maker

INTO manufacturer

FROM PRODUCT

WHERE Modelno = (

SELECT Modelno

FROM LAPTOP

ORDER BY Price DESC

LIMIT 1

);

END//

DELIMITER ;

-- Usage:

CALL FindMostExpensiveLaptopManufacturer(@manufacturer);

SELECT @manufacturer;

Design an input form for entering LAPTOP data. Apply possible validations.

# RCPET’s IMRD, Shirpur

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

# Duration 3:00 Hrs. Max marks: 20

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:-

I) Queries and their output.

2) Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

'PRODUCT ( Maker, Modelno, Type )

PC ( Modelno, Speed, RAM, HD, CD, Price )

LAPTOP ( Modelno, Speed, RAM, HD, Price )

PRINTER ( Modelno, Color, Type, Price )

Details regarding Schemas

* PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.
* The value for Maker in Product table can be IBM, Compaq,etc.
* PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

* The values of any attributes should not be null.
* Product Type should one of these (PC, Laptop or Printer) **Queries:**

1. Find the different types of printers produced by Epson.
2. Find those hard disk sizes which occur in two or more PC's.
3. Write a trigger on LAPTOP table such that the minimum speed should be 1201\4Hz.
4. Demonstrate the use of cursor using PRODUCT table.

Design an input form for entering LAPTOP data. Apply possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

# Duration 3:00 Hrs. Max marks : 20

# Instructions:

* Read the slip carefully.

'• Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.

* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification)

1AT1ENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Pid)

ADMITTEDPATIENT (P\_code, Entry date, Discharge date, wardno, disease )

# Integrity Constraints:

* The values of any attributes should not be null.  Gender value should be M male) or F(female).
* Wardno should be less than 6.

-- Create tables

CREATE TABLE DOCTOR (

Did INT PRIMARY KEY,

Dname VARCHAR(100) NOT NULL,

Daddress VARCHAR(100) NOT NULL,

qualification VARCHAR(100) NOT NULL

);

CREATE TABLE PATIENTMASTER (

Pcode INT PRIMARY KEY,

Pname VARCHAR(100) NOT NULL,

Padd VARCHAR(100) NOT NULL,

age INT NOT NULL,

gender CHAR(1) NOT NULL,

bloodgroup VARCHAR(5) NOT NULL,

Pid INT NOT NULL

);

CREATE TABLE ADMITTEDPATIENT (

P\_code INT PRIMARY KEY,

Entry\_date DATE NOT NULL,

Discharge\_date DATE,

wardno INT NOT NULL,

disease VARCHAR(100) NOT NULL,

CONSTRAINT CHK\_Gender CHECK (gender IN ('M', 'F')),

CONSTRAINT CHK\_Wardno CHECK (wardno < 6)

);

-- Insert data into tables

INSERT INTO DOCTOR (Did, Dname, Daddress, qualification) VALUES

(1, 'Doctor 1', 'Address 1', 'MBBS'),

(2, 'Doctor 2', 'Address 2', 'MD'),

(3, 'Doctor 3', 'Address 3', 'MS'),

(4, 'Doctor 4', 'Address 4', 'MBBS'),

(5, 'Doctor 5', 'Address 5', 'MD');

INSERT INTO PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Pid) VALUES

(1, 'Patient 1', 'Address 1', 30, 'M', 'A', 1001),

(2, 'Patient 2', 'Address 2', 25, 'F', 'B', 1002),

(3, 'Patient 3', 'Address 3', 40, 'M', 'O', 1003),

(4, 'Patient 4', 'Address 4', 50, 'F', 'AB', 1004);

INSERT INTO ADMITTEDPATIENT (P\_code, Entry\_date, Discharge\_date, wardno, disease) VALUES

(1, '2012-03-01', '2012-03-10', 2, 'Fever'),

(2, '2012-03-03', '2012-03-15', 3, 'Infection'),

(3, '2012-03-05', '2012-03-20', 3, 'Fracture'),

(4, '2012-03-08', '2012-03-18', 1, 'Pneumonia');

-- Integrity Constraints

ALTER TABLE DOCTOR ADD CONSTRAINT CHK\_Doctor\_Did CHECK (Did > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_age CHECK (age > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_bloodgroup CHECK (bloodgroup IN ('A', 'B', 'AB', 'O'));

-- Queries

a) Find the details of doctors who are treating the patients of ward no 3.

```sql

SELECT D.Did, D.Dname, D.Daddress, D.qualification

FROM DOCTOR D

INNER JOIN ADMITTEDPATIENT AP ON D.Did = AP.P\_code

WHERE AP.wardno = 3;

**Queries:**

* 1. Find the details of doctors who are treating the patient of ward no 3.

**CREATE TRIGGER trg\_PatientMaster\_BloodGroup**

**BEFORE INSERT OR UPDATE ON PATIENTMASTER**

**FOR EACH ROW**

**BEGIN**

**IF NEW.bloodgroup NOT IN ('A', 'B', 'AB', 'O') THEN**

**SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Blood group should be A, B, AB, or O.';**

**END IF;**

**END;**

* 1. Write a trigger on **PATIENTMASTER** table such that the blood group should be A,B,AB or 0.

**SELECT \***

**FROM ADMITTEDPATIENT**

**WHERE Discharge\_date BETWEEN '2012-03-03' AND '2012-03-25';**

* 1. Find the details of patient who are discharge within the period

03/03/12 to 25/ 03/12

* 1. Write a procedure on ADMIFTEDPATIENT table such as to calculate bill of all discharged patients. The charges per day for a ward is WardNo. \* 100. e.g. For ward no 3 charges/day are 300Rs.

**DELIMITER //**

**CREATE PROCEDURE CalculateBill()**

**BEGIN**

**DECLARE totalBill DECIMAL(10, 2);**

**SELECT SUM(DATEDIFF(Discharge\_date, Entry\_date) \* wardno \* 100)**

**INTO totalBill**

**FROM ADMITTEDPATIENT**

**WHERE Discharge\_date IS NOT NULL;**

**SELECT totalBill;**

**END//**

**DELIMITER ;**

**-- Usage:**

**CALL CalculateBill();**

Create a data report for the details of Doctors. Report should also include the details of patients treated by that doctor.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* Read the Scheinas carefully before filling records iP the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:

I) Queries and their output.

2) Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification)

PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did) ADMITTEDPATIENT (Pcode, Entry date, Discharge\_date, wardno, disease )

Integrity Constraints:

* The values of any attributes should not be null.
* Gender value should be M (male) or F(female).  Wardno should be less than 6.

Queries:

* 1. Find the details of the doctors who are treating the patients of ward no 3 & display the result along with patient name & disease.
  2. Find the name of the disease by which maximum patients are suffering.
  3. Write a trigger on ADMITTEDPATIENT table such that the wardno value should be between 1-5.
  4. Write a procedure to give the details of patients who are admitted in the hospital for more than 5 days.

Create a input form for doctors. Apply all possible validations.

CREATE TABLE DOCTOR (

Did INT PRIMARY KEY,

Dname VARCHAR(100) NOT NULL,

Daddress VARCHAR(100) NOT NULL,

qualification VARCHAR(100) NOT NULL

);

CREATE TABLE PATIENTMASTER (

Pcode INT PRIMARY KEY,

Pname VARCHAR(100) NOT NULL,

Padd VARCHAR(100) NOT NULL,

age INT NOT NULL,

gender CHAR(1) NOT NULL,

bloodgroup VARCHAR(5) NOT NULL,

Did INT NOT NULL,

CONSTRAINT CHK\_Gender CHECK (gender IN ('M', 'F')),

CONSTRAINT FK\_PatientMaster\_Doctor FOREIGN KEY (Did) REFERENCES DOCTOR(Did)

);

CREATE TABLE ADMITTEDPATIENT (

Pcode INT PRIMARY KEY,

Entry\_date DATE NOT NULL,

Discharge\_date DATE,

wardno INT NOT NULL,

disease VARCHAR(100) NOT NULL,

CONSTRAINT CHK\_Wardno CHECK (wardno BETWEEN 1 AND 5),

CONSTRAINT FK\_AdmittedPatient\_PatientMaster FOREIGN KEY (Pcode) REFERENCES PATIENTMASTER(Pcode)

);

-- Insert data into tables

INSERT INTO DOCTOR (Did, Dname, Daddress, qualification) VALUES

(1, 'Doctor 1', 'Address 1', 'MBBS'),

(2, 'Doctor 2', 'Address 2', 'MD'),

(3, 'Doctor 3', 'Address 3', 'MS'),

(4, 'Doctor 4', 'Address 4', 'MBBS'),

(5, 'Doctor 5', 'Address 5', 'MD');

INSERT INTO PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did) VALUES

(1, 'Patient 1', 'Address 1', 30, 'M', 'A', 1),

(2, 'Patient 2', 'Address 2', 25, 'F', 'B', 2),

(3, 'Patient 3', 'Address 3', 40, 'M', 'O', 3),

(4, 'Patient 4', 'Address 4', 50, 'F', 'AB', 4);

INSERT INTO ADMITTEDPATIENT (Pcode, Entry\_date, Discharge\_date, wardno, disease) VALUES

(1, '2023-01-01', '2023-01-05', 2, 'Fever'),

(2, '2023-01-02', '2023-01-07', 3, 'Infection'),

(3, '2023-01-03', '2023-01-08', 3, 'Fracture'),

(4, '2023-01-04', '2023-01-09', 1, 'Pneumonia');

-- Integrity Constraints

ALTER TABLE DOCTOR ADD CONSTRAINT CHK\_Doctor\_Did CHECK (Did > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_age CHECK (age > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_bloodgroup CHECK (bloodgroup IN ('A', 'B', 'AB', 'O'));

-- Queries

a) Find the details of the doctors who are treating the patients of ward no 3 and display the result along with patient name and disease.

```sql

SELECT D.Did, D.Dname, D.Daddress, P.Pname, AP.disease

FROM DOCTOR D

JOIN PATIENTMASTER P ON D.Did = P.Did

JOIN ADMITTEDPATIENT AP ON P.Pcode = AP.Pcode

WHERE AP.wardno = 3;

b) Find the name of the disease by which maximum patients are suffering.

sql

Copy code

SELECT disease

FROM ADMITTEDPATIENT

GROUP BY disease

ORDER BY COUNT(\*) DESC

LIMIT 1;

c) Write a trigger on ADMITTEDPATIENT table such that the wardno value should be between 1-5.

sql

Copy code

CREATE TRIGGER trg\_AdmittedPatient\_Wardno

BEFORE INSERT OR UPDATE ON ADMITTEDPATIENT

FOR EACH ROW

BEGIN

IF NEW.wardno NOT BETWEEN 1 AND 5 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Wardno value should be between 1 and 5.';

END IF;

END;

d) Write a procedure to give the details of patients who are admitted in the hospital for more than 5 days.

sql

Copy code

DELIMITER //

CREATE PROCEDURE FindLongTermPatients()

BEGIN

SELECT P.Pcode, P.Pname, AP.Entry\_date, AP.Discharge\_date

FROM PATIENTMASTER P

JOIN ADMITTEDPATIENT AP ON P.Pcode = AP.Pcode

WHERE DATEDIFF(AP.Discharge\_date, AP.Entry\_date) > 5;

END//

DELIMITER ;

-- Usage:

CALL FindLongTermPatients();

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS Duration 3:00 Hrs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion,cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least l 0 records in each table).

DOCTOR (Did, Dname, Daddress, qualification)

PATIENTMASTER (Pcode, Pnarne, Padd, age, gender, bloodgroup, aid) ADMITTEDPATIENT (Pcode, Entry\_date, Discharge date, wardno, disease )

Integrity Constraints:

* The values of any attributes should not be null.
* Gender value should be M (male) or F(female).
* Wardno should be less than 6. Queries:
  1. Find details of the patients who are treated by M.B.B.S. doctors.
  2. Find the details of patient who is suffered from blood cancer having age less than 50 years & blood group is A.
  3. write a procedure on ADMITTEDPATIENT table such as to calculate the bill of all patients. (bill no of days \* 600)
  4. Write a cursor on PATIENTMASTER table to fetch the last record & display the rows in that table.

Create a data entry for New Doctor's entry. Apply all possible validations

-- Create tables

CREATE TABLE DOCTOR (

Did INT PRIMARY KEY,

Dname VARCHAR(100) NOT NULL,

Daddress VARCHAR(100) NOT NULL,

qualification VARCHAR(100) NOT NULL

);

CREATE TABLE PATIENTMASTER (

Pcode INT PRIMARY KEY,

Pname VARCHAR(100) NOT NULL,

Padd VARCHAR(100) NOT NULL,

age INT NOT NULL,

gender CHAR(1) NOT NULL,

bloodgroup VARCHAR(5) NOT NULL,

aid INT NOT NULL,

CONSTRAINT CHK\_Gender CHECK (gender IN ('M', 'F')),

CONSTRAINT FK\_PatientMaster\_Doctor FOREIGN KEY (aid) REFERENCES DOCTOR(Did)

);

CREATE TABLE ADMITTEDPATIENT (

Pcode INT PRIMARY KEY,

Entry\_date DATE NOT NULL,

Discharge\_date DATE,

wardno INT NOT NULL,

disease VARCHAR(100) NOT NULL,

CONSTRAINT CHK\_Wardno CHECK (wardno < 6),

CONSTRAINT FK\_AdmittedPatient\_PatientMaster FOREIGN KEY (Pcode) REFERENCES PATIENTMASTER(Pcode)

);

-- Insert data into tables

INSERT INTO DOCTOR (Did, Dname, Daddress, qualification) VALUES

(1, 'Doctor 1', 'Address 1', 'MBBS'),

(2, 'Doctor 2', 'Address 2', 'MD'),

(3, 'Doctor 3', 'Address 3', 'MBBS'),

(4, 'Doctor 4', 'Address 4', 'MD'),

(5, 'Doctor 5', 'Address 5', 'MBBS');

INSERT INTO PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, aid) VALUES

(1, 'Patient 1', 'Address 1', 30, 'M', 'A', 1),

(2, 'Patient 2', 'Address 2', 40, 'F', 'B', 2),

(3, 'Patient 3', 'Address 3', 35, 'M', 'A', 3),

(4, 'Patient 4', 'Address 4', 45, 'F', 'AB', 4);

INSERT INTO ADMITTEDPATIENT (Pcode, Entry\_date, Discharge\_date, wardno, disease) VALUES

(1, '2023-01-01', '2023-01-05', 2, 'Fever'),

(2, '2023-01-02', '2023-01-07', 3, 'Blood Cancer'),

(3, '2023-01-03', '2023-01-08', 3, 'Fracture'),

(4, '2023-01-04', '2023-01-09', 1, 'Pneumonia');

-- Integrity Constraints

ALTER TABLE DOCTOR ADD CONSTRAINT CHK\_Doctor\_Did CHECK (Did > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_age CHECK (age > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_bloodgroup CHECK (bloodgroup IN ('A', 'B', 'AB', 'O'));

-- Queries

a) Find details of the patients who are treated by M.B.B.S. doctors.

```sql

SELECT PM.Pcode, PM.Pname, PM.Padd, PM.age, PM.gender, PM.bloodgroup

FROM PATIENTMASTER PM

JOIN DOCTOR D ON PM.aid = D.Did

WHERE D.qualification = 'MBBS';

b) Find the details of the patient who has suffered from blood cancer, is less than 50 years old, and has blood group A.

sql

Copy code

SELECT \*

FROM PATIENTMASTER

WHERE disease = 'Blood Cancer' AND age < 50 AND bloodgroup = 'A';

c) Write a procedure on ADMITTEDPATIENT table to calculate the bill of all patients. (bill = no of days \* 600)

sql

Copy code

DELIMITER //

CREATE PROCEDURE CalculateBill()

BEGIN

DECLARE totalBill DECIMAL(10, 2);

SELECT SUM(DATEDIFF(Discharge\_date, Entry\_date) \* 600)

INTO totalBill

FROM ADMITTEDPATIENT

WHERE Discharge\_date IS NOT NULL;

SELECT totalBill;

END//

DELIMITER ;

-- Usage:

CALL CalculateBill();

d) Write a cursor on PATIENTMASTER table to fetch the last record and display the rows in that table.

sql

Copy code

DECLARE cur CURSOR FOR SELECT \* FROM PATIENTMASTER;

DECLARE @Pcode INT, @Pname VARCHAR(100), @Padd VARCHAR(100), @age INT, @gender CHAR(1), @bloodgroup VARCHAR(5), @aid INT;

OPEN cur;

FETCH LAST FROM cur INTO @Pcode, @Pname, @Padd, @age, @gender, @bloodgroup, @aid;

WHILE @@FETCH\_STATUS = 0

BEGIN

-- Display row data

PRINT 'Pcode: ' + CONVERT(VARCHAR(10), @Pcode);

PRINT 'Pname: ' + @Pname;

PRINT 'Padd: ' +

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification)

PATIENTMASTER (Pcocle, Pname, Padd, age, gender, bloodgroup, Did)

ADMITTEDPATIENT (Pcode, Entry date, Discharge date. wardno, disease )

**Integrity Constraints:**

* The values of any attributes should not be null.
* Gender value should be M (male) or F(female).
* Wardno should be less than 6.
* **Queries:** 
  1. Find details of the patients who are treated by M.S. doctors.
  2. Find the name of doctor who is treating maximum number of patients.
  3. Write a procedure to give the details of patients who are admitted in the hospital for more than 15 days.
  4. Create a view on DOCTOR & PATIENTMASTER tables. Update details of the patients who are treated by B.A.-M.S. doctors **to M.B.B.S** doctor.

Create a data entry form for discharging a patient. Also give information regarding his bill. (bill no\_of\_days \* 500)

CREATE TABLE DOCTOR (

Did INT PRIMARY KEY,

Dname VARCHAR(100) NOT NULL,

Daddress VARCHAR(100) NOT NULL,

qualification VARCHAR(100) NOT NULL

);

CREATE TABLE PATIENTMASTER (

Pcode INT PRIMARY KEY,

Pname VARCHAR(100) NOT NULL,

Padd VARCHAR(100) NOT NULL,

age INT NOT NULL,

gender CHAR(1) NOT NULL,

bloodgroup VARCHAR(5) NOT NULL,

Did INT NOT NULL,

CONSTRAINT CHK\_Gender CHECK (gender IN ('M', 'F')),

CONSTRAINT FK\_PatientMaster\_Doctor FOREIGN KEY (Did) REFERENCES DOCTOR(Did)

);

CREATE TABLE ADMITTEDPATIENT (

Pcode INT PRIMARY KEY,

Entry\_date DATE NOT NULL,

Discharge\_date DATE,

wardno INT NOT NULL,

disease VARCHAR(100) NOT NULL,

CONSTRAINT CHK\_Wardno CHECK (wardno < 6),

CONSTRAINT FK\_AdmittedPatient\_PatientMaster FOREIGN KEY (Pcode) REFERENCES PATIENTMASTER(Pcode)

);

-- Insert data into tables

INSERT INTO DOCTOR (Did, Dname, Daddress, qualification) VALUES

(1, 'Doctor 1', 'Address 1', 'M.S.'),

(2, 'Doctor 2', 'Address 2', 'M.D.'),

(3, 'Doctor 3', 'Address 3', 'M.S.'),

(4, 'Doctor 4', 'Address 4', 'M.D.'),

(5, 'Doctor 5', 'Address 5', 'M.S.');

INSERT INTO PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did) VALUES

(1, 'Patient 1', 'Address 1', 30, 'M', 'A', 1),

(2, 'Patient 2', 'Address 2', 40, 'F', 'B', 2),

(3, 'Patient 3', 'Address 3', 35, 'M', 'A', 3),

(4, 'Patient 4', 'Address 4', 45, 'F', 'AB', 4);

INSERT INTO ADMITTEDPATIENT (Pcode, Entry\_date, Discharge\_date, wardno, disease) VALUES

(1, '2023-01-01', '2023-01-05', 2, 'Fever'),

(2, '2023-01-02', '2023-01-07', 3, 'Blood Cancer'),

(3, '2023-01-03', '2023-01-08', 3, 'Fracture'),

(4, '2023-01-04', '2023-01-09', 1, 'Pneumonia');

-- Integrity Constraints

ALTER TABLE DOCTOR ADD CONSTRAINT CHK\_Doctor\_Did CHECK (Did > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_age CHECK (age > 0);

ALTER TABLE PATIENTMASTER ADD CONSTRAINT CHK\_PatientMaster\_bloodgroup CHECK (bloodgroup IN ('A', 'B', 'AB', 'O'));

-- Queries

a) Find details of the patients who are treated by M.S. doctors.

```sql

SELECT PM.Pcode, PM.Pname, PM.Padd, PM.age, PM.gender, PM.bloodgroup

FROM PATIENTMASTER PM

JOIN DOCTOR D ON PM.Did = D.Did

WHERE D.qualification = 'M.S.';

b) Find the name of the doctor who is treating the maximum number of patients.

sql

Copy code

SELECT D.Dname

FROM DOCTOR D

JOIN PATIENTMASTER PM ON D.Did = PM.Did

GROUP BY D.Dname

ORDER BY COUNT(\*) DESC

LIMIT 1;

c) Write a procedure to give the details of patients who are admitted in the hospital for more than 15 days.

sql

Copy code

DELIMITER //

CREATE PROCEDURE FindLongTermPatients()

BEGIN

SELECT P.\*

FROM PATIENTMASTER P

JOIN ADMITTEDPATIENT AP ON P.Pcode = AP.Pcode

WHERE DATEDIFF(AP.Discharge\_date, AP.Entry\_date) > 15;

END//

DELIMITER ;

-- Usage:

CALL FindLongTermPatients();

d) Create a view on DOCTOR & PATIENTMASTER tables. Update details of the patients who are treated by B.A.-M.S. doctors to M.B.B.S doctors.

sql

Copy code

-- Create a view

CREATE VIEW DoctorPatientView AS

SELECT D.Did, D.Dname, D.Daddress, D.qualification, PM.Pcode, PM.Pname, PM.Padd, PM.age, PM.gender, PM.bloodgroup

FROM DOCTOR D

JOIN PATIENTMASTER PM ON D.Did = PM.Did;

-- Update patients treated by B.A.-M.S. doctors to M.B.B.S doctors

UPDATE PATIENTMASTER

SET Did = (

SELECT Did

FROM DOCTOR

WHERE qualification = 'M.B.B.S.'

)

WHERE Did IN (

SELECT Did

FROM DOCTOR

WHERE qualification = 'B.A.-M.S.'

);

To create a data entry form for discharging a patient and calculate their bill (bill = no\_of\_days \* 500), you can use a programming language or framework to create a user interface that interacts with the database. The specific design and implementation would depend on the technology stack you c

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before tilling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

ACCOUNT (accno, open date. acctype, balance)

TRANSACTION (trans id, trans date, accno, trans type, amount) CUSTOMER

(cust id, name, address, Accno)

Integrity Constraints:

* The values of any attributes should not be null.
* acctype value should be P(Personal) or J(Joint).
* Accno should be less than 3 digits.
* Trans type should be C(Credit) or D(Debit)

Queries:

1. Find the details of customers whose minimum balance is 1 lakhs.
2. Find the details of amount credited within the period 25-3-2012 to 28-3- 2012
3. Write a trigger on TRANSACTION table to calculate current balance of account on which transaction is made.
4. Write a cursor on ACCOUNT table of balance attribute such that if the balance is less than 10000 then print the 'loan is not provided'else 'loan is provided'.

Create a data entry for New account entry. Apply all possible validations

CREATE TABLE ACCOUNT (

accno INT PRIMARY KEY,

open\_date DATE NOT NULL,

acctype CHAR(1) NOT NULL,

balance DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_Accno CHECK (accno < 100),

CONSTRAINT CHK\_Acctype CHECK (acctype IN ('P', 'J'))

);

CREATE TABLE TRANSACTION (

trans\_id INT PRIMARY KEY,

trans\_date DATE NOT NULL,

accno INT NOT NULL,

trans\_type CHAR(1) NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_TransType CHECK (trans\_type IN ('C', 'D')),

CONSTRAINT FK\_Transaction\_Account FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)

);

CREATE TABLE CUSTOMER (

cust\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

address VARCHAR(100) NOT NULL,

Accno INT NOT NULL,

CONSTRAINT FK\_Customer\_Account FOREIGN KEY (Accno) REFERENCES ACCOUNT(accno)

);

-- Insert data into tables

INSERT INTO ACCOUNT (accno, open\_date, acctype, balance) VALUES

(1, '2023-01-01', 'P', 50000),

(2, '2023-01-02', 'J', 100000),

(3, '2023-01-03', 'P', 150000),

(4, '2023-01-04', 'J', 200000);

INSERT INTO TRANSACTION (trans\_id, trans\_date, accno, trans\_type, amount) VALUES

(1, '2023-01-01', 1, 'C', 10000),

(2, '2023-01-02', 1, 'D', 5000),

(3, '2023-01-03', 2, 'C', 20000),

(4, '2023-01-04', 2, 'D', 10000);

INSERT INTO CUSTOMER (cust\_id, name, address, Accno) VALUES

(1, 'Customer 1', 'Address 1', 1),

(2, 'Customer 2', 'Address 2', 2),

(3, 'Customer 3', 'Address 3', 3),

(4, 'Customer 4', 'Address 4', 4);

-- Integrity Constraints

ALTER TABLE ACCOUNT ADD CONSTRAINT CHK\_Account\_Balance CHECK (balance >= 0);

-- Queries

a) Find the details of customers whose minimum balance is 1 lakh.

```sql

SELECT C.cust\_id, C.name, C.address, A.accno, A.balance

FROM CUSTOMER C

JOIN ACCOUNT A ON C.Accno = A.accno

WHERE A.balance >= 100000;

b) Find the details of amounts credited within the period 25-3-2012 to 28-3-2012.

sql

Copy code

SELECT T.trans\_id, T.trans\_date, T.accno, T.amount

FROM TRANSACTION T

WHERE T.trans\_type = 'C'

AND T.trans\_date BETWEEN '2012-03-25' AND '2012-03-28';

c) Write a trigger on TRANSACTION table to calculate the current balance of the account on which the transaction is made.

sql

Copy code

CREATE TRIGGER trg\_Transaction\_UpdateBalance

AFTER INSERT ON TRANSACTION

FOR EACH ROW

BEGIN

DECLARE currBalance DECIMAL(10, 2);

SELECT balance INTO currBalance

FROM ACCOUNT

WHERE accno = NEW.accno;

IF NEW.trans\_type = 'C' THEN

SET currBalance = currBalance + NEW.amount;

ELSE

SET currBalance = currBalance - NEW.amount;

END IF;

UPDATE ACCOUNT

SET balance = currBalance

WHERE accno = NEW.accno;

END;

d) Write a cursor on ACCOUNT table for the balance attribute such that if the balance is less than 10000, then print 'loan is not provided'; else print 'loan is provided'.

sql

Copy code

DECLARE @accno INT, @balance DECIMAL(10, 2);

DECLARE cur CURSOR FOR SELECT accno, balance FROM ACCOUNT;

OPEN cur;

FETCH NEXT FROM cur INTO @accno, @balance;

WHILE @@FETCH\_STATUS = 0

BEGIN

IF @balance < 10000 THEN

PRINT 'Loan is not provided for Account: ' + CONVERT(VARCHAR(10), @accno);

ELSE

PRINT 'Loan is provided for Account: ' + CONVERT(VARCHAR(10), @accno);

END IF;

FETCH NEXT FROM cur INTO @accno, @balance;

END;

CLOSE cur;

DEALLOCATE cur;

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS Duration 3:00 Hrs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* .Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of
  1. Queries and their output..
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

ACCOUNT (accno, open date, acctype, balance)

TRANSACTION (trans id, trans date, cno, trans type, amount) CUSTOMER (cust id, name, address, accno)

Integrity Constraints:

* The values of any attributes should not be null.
* acctype value should be P(Personal) or J(Joint).
* Accno should be less than 3 digits.
* Trans\_type should be **C(Credit) Or** D(Debit)

Queries:

* 1. Find the details of customers who have personal accounts & balance is less than 2 lakhs.
  2. Find the details of customers who have joint accounts.
  3. Write a trigger on ACCOUNT table such that if balance is less than 300 then customer should not withdraw the money.
  4. Write a procedure on ACCOUNT & TRANSACTION table such that as user enters new transaction the balance is, updated in ACCOUNT table.

Create a data entry for New customer entry. Apply all possible validations.

CREATE TABLE ACCOUNT (

accno INT PRIMARY KEY,

open\_date DATE NOT NULL,

acctype CHAR(1) NOT NULL,

balance DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_Accno CHECK (accno < 100),

CONSTRAINT CHK\_Acctype CHECK (acctype IN ('P', 'J'))

);

CREATE TABLE TRANSACTION (

trans\_id INT PRIMARY KEY,

trans\_date DATE NOT NULL,

accno INT NOT NULL,

trans\_type CHAR(1) NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_TransType CHECK (trans\_type IN ('C', 'D')),

CONSTRAINT FK\_Transaction\_Account FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)

);

CREATE TABLE CUSTOMER (

cust\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

address VARCHAR(100) NOT NULL,

accno INT NOT NULL,

CONSTRAINT FK\_Customer\_Account FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)

);

-- Insert data into tables

INSERT INTO ACCOUNT (accno, open\_date, acctype, balance) VALUES

(1, '2023-01-01', 'P', 150000),

(2, '2023-01-02', 'J', 250000),

(3, '2023-01-03', 'P', 180000),

(4, '2023-01-04', 'J', 350000);

INSERT INTO TRANSACTION (trans\_id, trans\_date, accno, trans\_type, amount) VALUES

(1, '2023-01-01', 1, 'C', 10000),

(2, '2023-01-02', 1, 'D', 5000),

(3, '2023-01-03', 2, 'C', 20000),

(4, '2023-01-04', 2, 'D', 10000);

INSERT INTO CUSTOMER (cust\_id, name, address, accno) VALUES

(1, 'Customer 1', 'Address 1', 1),

(2, 'Customer 2', 'Address 2', 2),

(3, 'Customer 3', 'Address 3', 3),

(4, 'Customer 4', 'Address 4', 4);

-- Integrity Constraints

ALTER TABLE ACCOUNT ADD CONSTRAINT CHK\_Account\_Balance CHECK (balance >= 0);

-- Queries

a) Find the details of customers who have personal accounts and balance is less than 2 lakhs.

```sql

SELECT C.cust\_id, C.name, C.address, A.accno, A.balance

FROM CUSTOMER C

JOIN ACCOUNT A ON C.accno = A.accno

WHERE A.acctype = 'P' AND A.balance < 200000;

b) Find the details of customers who have joint accounts.

sql

Copy code

SELECT C.cust\_id, C.name, C.address, A.accno

FROM CUSTOMER C

JOIN ACCOUNT A ON C.accno = A.accno

WHERE A.acctype = 'J';

c) Write a trigger on ACCOUNT table such that if the balance is less than 300, the customer should not withdraw the money.

sql

Copy code

CREATE TRIGGER trg\_Account\_CheckBalance

BEFORE UPDATE ON ACCOUNT

FOR EACH ROW

BEGIN

IF NEW.balance < 300 AND NEW.acctype = 'P' THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Cannot withdraw money when the balance is less than 300.';

END IF;

END;

d) Write a procedure on ACCOUNT and TRANSACTION table such that as the user enters a new transaction, the balance is updated in the ACCOUNT table.

sql

Copy code

DELIMITER //

CREATE PROCEDURE UpdateAccountBalance(IN trans\_id INT, IN trans\_date DATE, IN accno INT, IN trans\_type CHAR(1), IN amount DECIMAL(10, 2))

BEGIN

DECLARE currBalance DECIMAL(10, 2);

-- Get the current balance

SELECT balance INTO currBalance

FROM ACCOUNT

WHERE accno = accno;

-- Update the balance based on transaction type

IF trans\_type = 'C' THEN

SET currBalance = currBalance + amount;

ELSE

SET currBalance = currBalance - amount;

END IF;

-- Update the balance in ACCOUNT table

UPDATE ACCOUNT

SET balance = currBalance

WHERE accno = accno;

-- Insert the transaction into TRANSACTION table

INSERT INTO TRANSACTION (trans\_id, trans\_date, accno, trans\_type, amount)

VALUES (trans\_id, trans\_date, accno, trans\_type, amount);

END//

DELIMITER ;

-- Usage:

CALL UpdateAccountBalance(5, '2023-01-05', 1, 'D', 1000);

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

# CA-LAB I (NEW) : Lab on DBMS Duration 3:00 Hrs. Max marks : 20

# Instructions:

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

ACCOUNT (accno, open date. acctype, balance)

TRANSACTION (trans id, trans date, accno, trans type, amount) CUSTOMER (cust id, name, address. accno)

Integrity Constraints:

* The values of any attributes should not be null.
* acctype value should be P(Personal) or Moira):
* Accno should be less than 3 digits.
* Transtype should be C(Credit) or D(Debit)

Queries:

* 1. Find the details of all transactions performed on account number 101. Also specify the name/names of cutomers who owns that account.
  2. Find the details ol amount credited within the period 15 -3-2012 to 18 -3 -2012.
  3. Write a trigger on insert on ACCOUNT table such that the account which is having balance less than or equal to 500 should not be debited.
  4. Write a procedure on ACCOUNT table to calculate interest on current balance from open\_date to today's date. (Take interest rate from user).

Create a data entry for New account entry. Apply all possible validations.

CREATE TABLE ACCOUNT (

accno INT PRIMARY KEY,

open\_date DATE NOT NULL,

acctype CHAR(1) NOT NULL,

balance DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_Accno CHECK (accno < 100),

CONSTRAINT CHK\_Acctype CHECK (acctype IN ('P', 'J'))

);

CREATE TABLE TRANSACTION (

trans\_id INT PRIMARY KEY,

trans\_date DATE NOT NULL,

accno INT NOT NULL,

trans\_type CHAR(1) NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_TransType CHECK (trans\_type IN ('C', 'D')),

CONSTRAINT FK\_Transaction\_Account FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)

);

CREATE TABLE CUSTOMER (

cust\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

address VARCHAR(100) NOT NULL,

accno INT NOT NULL,

CONSTRAINT FK\_Customer\_Account FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)

);

-- Insert data into tables

INSERT INTO ACCOUNT (accno, open\_date, acctype, balance) VALUES

(1, '2023-01-01', 'P', 150000),

(2, '2023-01-02', 'J', 250000),

(3, '2023-01-03', 'P', 180000),

(4, '2023-01-04', 'J', 350000);

INSERT INTO TRANSACTION (trans\_id, trans\_date, accno, trans\_type, amount) VALUES

(1, '2023-01-01', 1, 'C', 10000),

(2, '2023-01-02', 1, 'D', 5000),

(3, '2023-01-03', 2, 'C', 20000),

(4, '2023-01-04', 2, 'D', 10000);

INSERT INTO CUSTOMER (cust\_id, name, address, accno) VALUES

(1, 'Customer 1', 'Address 1', 1),

(2, 'Customer 2', 'Address 2', 2),

(3, 'Customer 3', 'Address 3', 3),

(4, 'Customer 4', 'Address 4', 4);

-- Integrity Constraints

ALTER TABLE ACCOUNT ADD CONSTRAINT CHK\_Account\_Balance CHECK (balance >= 0);

-- Queries

a) Find the details of customers who have personal accounts and balance is less than 2 lakhs.

```sql

SELECT C.cust\_id, C.name, C.address, A.accno, A.balance

FROM CUSTOMER C

JOIN ACCOUNT A ON C.accno = A.accno

WHERE A.acctype = 'P' AND A.balance < 200000;

b) Find the details of customers who have joint accounts.

sql

Copy code

SELECT C.cust\_id, C.name, C.address, A.accno

FROM CUSTOMER C

JOIN ACCOUNT A ON C.accno = A.accno

WHERE A.acctype = 'J';

c) Write a trigger on ACCOUNT table such that if the balance is less than 300, the customer should not withdraw the money.

sql

Copy code

CREATE TRIGGER trg\_Account\_CheckBalance

BEFORE UPDATE ON ACCOUNT

FOR EACH ROW

BEGIN

IF NEW.balance < 300 AND NEW.acctype = 'P' THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Cannot withdraw money when the balance is less than 300.';

END IF;

END;

d) Write a procedure on ACCOUNT and TRANSACTION table such that as the user enters a new transaction, the balance is updated in the ACCOUNT table.

sql

Copy code

DELIMITER //

CREATE PROCEDURE UpdateAccountBalance(IN trans\_id INT, IN trans\_date DATE, IN accno INT, IN trans\_type CHAR(1), IN amount DECIMAL(10, 2))

BEGIN

DECLARE currBalance DECIMAL(10, 2);

-- Get the current balance

SELECT balance INTO currBalance

FROM ACCOUNT

WHERE accno = accno;

-- Update the balance based on transaction type

IF trans\_type = 'C' THEN

SET currBalance = currBalance + amount;

ELSE

SET currBalance = currBalance - amount;

END IF;

-- Update the balance in ACCOUNT table

UPDATE ACCOUNT

SET balance = currBalance

WHERE accno = accno;

-- Insert the transaction into TRANSACTION table

INSERT INTO TRANSACTION (trans\_id, trans\_date, accno, trans\_type, amount)

VALUES (trans\_id, trans\_date, accno, trans\_type, amount);

END//

DELIMITER ;

-- Usage:

CALL UpdateAccountBalance(5, '2023-01-05', 1, 'D', 1000);

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

# Duration 3:00 Hrs. CA-LAB I (NEW) : Lab on DBMS

**Max marks : 20**

# Instructions:

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately gothat mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:

I) Queries and their output.

2) Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

.Creat4 database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

ACCOUNT (accno, open date, acctype, balance)

TRANSACTION (trans id, trans date, accno, trans type, amount)

CUSTOMER (cust id name, address, accno) Integrity Constraints:

* The values of any attributes should not be null.
* acctype value should be P(Personal) or Moira).  Accno should be less than 3 digits.
* Trans\_type should be C(Credit) or D(Debit)

Queries:

* 1. Find the details of customers who have opened the accounts within the period 25-3-2012 to 28-3-2012.
  2. Find the details of customers who have joint accounts & balance is less than 2 lakhs.
  3. Write a trigger TRANSACTION on table to calculate the current balance of the account on which transaction is made.

( if trans\_type = c then bal = bal amt else if trans\_type = d then bal = bal — amt)

* 1. write a cursor on CUSTOMER table to fetch the last row.

Create a data entry for New customer entry. Apply all possible validations.

CREATE TABLE ACCOUNT (

accno INT PRIMARY KEY,

open\_date DATE NOT NULL,

acctype CHAR(1) NOT NULL,

balance DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_Accno CHECK (accno < 100),

CONSTRAINT CHK\_Acctype CHECK (acctype IN ('P', 'M'))

);

CREATE TABLE TRANSACTION (

trans\_id INT PRIMARY KEY,

trans\_date DATE NOT NULL,

accno INT NOT NULL,

trans\_type CHAR(1) NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

CONSTRAINT CHK\_TransType CHECK (trans\_type IN ('C', 'D')),

CONSTRAINT FK\_Transaction\_Account FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)

);

CREATE TABLE CUSTOMER (

cust\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

address VARCHAR(100) NOT NULL,

accno INT NOT NULL,

CONSTRAINT FK\_Customer\_Account FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)

);

-- Insert data into tables

INSERT INTO ACCOUNT (accno, open\_date, acctype, balance) VALUES

(1, '2023-01-01', 'P', 100000),

(2, '2023-01-02', 'M', 200000),

(3, '2023-01-03', 'P', 150000),

(4, '2023-01-04', 'M', 250000);

INSERT INTO TRANSACTION (trans\_id, trans\_date, accno, trans\_type, amount) VALUES

(1, '2023-01-01', 1, 'C', 10000),

(2, '2023-01-02', 1, 'D', 5000),

(3, '2023-01-03', 2, 'C', 20000),

(4, '2023-01-04', 2, 'D', 10000);

INSERT INTO CUSTOMER (cust\_id, name, address, accno) VALUES

(1, 'Customer 1', 'Address 1', 1),

(2, 'Customer 2', 'Address 2', 2),

(3, 'Customer 3', 'Address 3', 3),

(4, 'Customer 4', 'Address 4', 4);

-- Integrity Constraints

ALTER TABLE ACCOUNT ADD CONSTRAINT CHK\_Account\_Balance CHECK (balance >= 0);

-- Queries

a) Find the details of customers who have opened accounts within the period 25-3-2012 to 28-3-2012.

```sql

SELECT C.cust\_id, C.name, C.address, A.accno, A.open\_date

FROM CUSTOMER C

JOIN ACCOUNT A ON C.accno = A.accno

WHERE A.open\_date BETWEEN '2012-03-25' AND '2012-03-28';

b) Find the details of customers who have joint accounts and balance is less than 2 lakhs.

sql

Copy code

SELECT C.cust\_id, C.name, C.address, A.accno, A.balance

FROM CUSTOMER C

JOIN ACCOUNT A ON C.accno = A.accno

WHERE A.acctype = 'M' AND A.balance < 200000;

c) Write a trigger on TRANSACTION table to calculate the current balance of the account on which the transaction is made. (if trans\_type = 'C' then bal = bal + amt, else if trans\_type = 'D' then bal = bal - amt)

sql

Copy code

CREATE TRIGGER trg\_Transaction\_UpdateBalance

AFTER INSERT ON TRANSACTION

FOR EACH ROW

BEGIN

IF NEW.trans\_type = 'C' THEN

UPDATE ACCOUNT SET balance = balance + NEW.amount WHERE accno = NEW.accno;

ELSEIF NEW.trans\_type = 'D' THEN

UPDATE ACCOUNT SET balance = balance - NEW.amount WHERE accno = NEW.accno;

END IF;

END;

d) Write a cursor on CUSTOMER table to fetch the last row.

sql

Copy code

DECLARE @cust\_id INT, @name VARCHAR(100), @address VARCHAR(100), @accno INT;

DECLARE cur CURSOR FOR SELECT cust\_id, name, address, accno FROM CUSTOMER;

OPEN cur;

FETCH LAST FROM cur INTO @cust\_id, @name, @address, @accno;

PRINT 'Last Customer: cust\_id=' + CONVERT(VARCHAR(10), @cust\_id) + ', name=' + @name + ', address=' + @address + ', accno=' + CONVERT(VARCHAR(10), @accno);

CLOSE cur;

DEALLOCATE cur;

**North Maharashtra University, Jalgaon**

**Internal PrNorth Maharashtra LIniversity, Jalgaon.ctical Exam nation Feb-2023**

# CA-LAB I (NEW) : Lab on DBMS Duration 3:00 Hrs. Max marks : 20

#  Instructions:

* Read the slip carefully.
* R-ead-theSchernas.vatefuttyliefore. finitig'retordsinlhe tableikftittlyerecords appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  + 1. Queries and their output.
    2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in eae,h table).

EMPLOYEE ( fname, Mame, ssn, sex, salary, joindate,superssn, dno,)

DEPT ( dname, dnum, mgrssn, dlocation)

PROJECT ( pname, pno, plocation, dnumber) WORK ON ( ssn, pno, hours) Integrity Constraints:

* The values of any attributes should not be null.
* The deptno should less than 4 digit
* ssn -(social security no of-employee)
* mgrssn(manager\_social\_security\_no)
* superssn(supervisor\_social\_securit)'\_no)

Queries:

* 1. For every project located in 'jalgaon". List the pno,the controlling detptno and dept manager last name.
  2. For each project on which more than two employee work, Find the pno, pname & no. of employees who work on the project.
  3. create a view that has the deptname,manager name & manager salary for every dept.
  4. Express the following constraint as SQL assertions -

"salary of employee must not be greater than the salary of the manager of the dept".

Create a data entry for New employee entry. Apply all possible validations.

CREATE TABLE EMPLOYEE (

fname VARCHAR(100) NOT NULL,

lname VARCHAR(100) NOT NULL,

ssn INT PRIMARY KEY,

sex CHAR(1) NOT NULL,

salary DECIMAL(10, 2) NOT NULL,

joindate DATE NOT NULL,

superssn INT,

dno INT NOT NULL,

CONSTRAINT CHK\_SSN CHECK (ssn < 10000),

CONSTRAINT CHK\_Sex CHECK (sex IN ('M', 'F')),

CONSTRAINT FK\_Employee\_SuperSsn FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn),

CONSTRAINT FK\_Employee\_Dept FOREIGN KEY (dno) REFERENCES DEPT(dnum)

);

CREATE TABLE DEPT (

dname VARCHAR(100) NOT NULL,

dnum INT PRIMARY KEY,

mgrssn INT NOT NULL,

dlocation VARCHAR(100) NOT NULL,

CONSTRAINT CHK\_DeptNo CHECK (dnum < 10000),

CONSTRAINT FK\_Dept\_Employee FOREIGN KEY (mgrssn) REFERENCES EMPLOYEE(ssn)

);

CREATE TABLE PROJECT (

pname VARCHAR(100) NOT NULL,

pno INT PRIMARY KEY,

plocation VARCHAR(100) NOT NULL,

dnumber INT NOT NULL,

CONSTRAINT CHK\_ProjectNo CHECK (pno < 10000),

CONSTRAINT FK\_Project\_Dept FOREIGN KEY (dnumber) REFERENCES DEPT(dnum)

);

CREATE TABLE WORKON (

ssn INT NOT NULL,

pno INT NOT NULL,

hours INT NOT NULL,

CONSTRAINT FK\_WorkOn\_Employee FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn),

CONSTRAINT FK\_WorkOn\_Project FOREIGN KEY (pno) REFERENCES PROJECT(pno)

);

-- Integrity Constraints

ALTER TABLE EMPLOYEE ADD CONSTRAINT CHK\_Employee\_Salary CHECK (salary >= 0);

-- Queries

a) For every project located in 'jalgaon', list the pno, the controlling deptno, and dept manager last name.

```sql

SELECT P.pno, D.dnum AS controlling\_deptno, E.lname AS manager\_last\_name

FROM PROJECT P

JOIN DEPT D ON P.dnumber = D.dnum

JOIN EMPLOYEE E ON D.mgrssn = E.ssn

WHERE P.plocation = 'jalgaon';

b) For each project on which more than two employees work, find the pno, pname, and the number of employees who work on the project.

sql

Copy code

SELECT P.pno, P.pname, COUNT(W.ssn) AS num\_employees

FROM PROJECT P

JOIN WORKON W ON P.pno = W.pno

GROUP BY P.pno, P.pname

HAVING COUNT(W.ssn) > 2;

c) Create a view that has the deptname, manager name, and manager salary for every department.

sql

Copy code

CREATE VIEW DeptManagerView AS

SELECT D.dname AS deptname, E.fname || ' ' || E.lname AS manager\_name, E.salary AS manager\_salary

FROM DEPT D

JOIN EMPLOYEE E ON D.mgrssn = E.ssn;

d) Express the following constraint as SQL assertions: "salary of employee must not be greater than the salary of the manager of the dept".

sql

Copy code

CREATE ASSERTION SalaryConstraint

CHECK (

NOT EXISTS (

SELECT \*

FROM EMPLOYEE E

JOIN DEPT D ON E.dno = D.dnum

WHERE E.salary > (SELECT M.salary FROM EMPLOYEE M WHERE M.ssn = D.mgrssn)

)

);

**North Maharashtra University, Jalgaon**

**Internal PrNorth Maharashtra LIniversity, Jalgaon.ctical Exam nation Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

# Duration 3:00 Hrs. Max marks : 20

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before tilling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion,cursor, views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

EMPLOYEE ( fname, lname, ssn, sex, salary, joindate,superssn, dno,)

DEPT ( dname, dnum, En\_grssn, (llocation)

PROJECT ( pname, pno, plocation, dnumber) WORK ON ( 5sn, pno, hours)

Integrity Constraints:

* The values of any attributes should not be null.
* The deptno should less than 4 digit
* Ssn (social security\_no of employee)
* Mgrssn(managersocial\_security\_no) a Superssn(supervisor\_social\_security\_no) Queries:

1. For each employee, Find the employee first & last name & the first & last name of his or her immediate supervisor.
2. For each dept, Find the deptno, the no. of employees in the dept & their average salary.
3. Create a view that has pname,controlling dept name, no of employees, & total hours worked per week on the project for each project with more than one employee working on it.
4. Create a procedure on EMPLOYEE table to determine the employees who will get promotion. (An employee will get promotion after working on 5 projects.)

Create a data report showing the information of all female employees working in "Research" department.

CREATE TABLE EMPLOYEE (

fname VARCHAR(100) NOT NULL,

lname VARCHAR(100) NOT NULL,

ssn INT PRIMARY KEY,

sex CHAR(1) NOT NULL,

salary DECIMAL(10, 2) NOT NULL,

joindate DATE NOT NULL,

superssn INT,

dno INT NOT NULL,

CONSTRAINT CHK\_SSN CHECK (ssn < 10000),

CONSTRAINT CHK\_Sex CHECK (sex IN ('M', 'F')),

CONSTRAINT FK\_Employee\_SuperSsn FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn),

CONSTRAINT FK\_Employee\_Dept FOREIGN KEY (dno) REFERENCES DEPT(dnum)

);

CREATE TABLE DEPT (

dname VARCHAR(100) NOT NULL,

dnum INT PRIMARY KEY,

mgrssn INT NOT NULL,

dlocation VARCHAR(100) NOT NULL,

CONSTRAINT CHK\_DeptNo CHECK (dnum < 10000),

CONSTRAINT FK\_Dept\_Employee FOREIGN KEY (mgrssn) REFERENCES EMPLOYEE(ssn)

);

CREATE TABLE PROJECT (

pname VARCHAR(100) NOT NULL,

pno INT PRIMARY KEY,

plocation VARCHAR(100) NOT NULL,

dnumber INT NOT NULL,

CONSTRAINT CHK\_ProjectNo CHECK (pno < 10000),

CONSTRAINT FK\_Project\_Dept FOREIGN KEY (dnumber) REFERENCES DEPT(dnum)

);

CREATE TABLE WORKON (

ssn INT NOT NULL,

pno INT NOT NULL,

hours INT NOT NULL,

CONSTRAINT FK\_WorkOn\_Employee FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn),

CONSTRAINT FK\_WorkOn\_Project FOREIGN KEY (pno) REFERENCES PROJECT(pno)

);

-- Integrity Constraints

ALTER TABLE EMPLOYEE ADD CONSTRAINT CHK\_Employee\_Salary CHECK (salary >= 0);

-- Queries

a) For each employee, find the employee's first & last name and the first & last name of their immediate supervisor.

```sql

SELECT E1.fname AS employee\_fname, E1.lname AS employee\_lname, E2.fname AS supervisor\_fname, E2.lname AS supervisor\_lname

FROM EMPLOYEE E1

LEFT JOIN EMPLOYEE E2 ON E1.superssn = E2.ssn;

b) For each department, find the department number, the number of employees in the department, and their average salary.

sql

Copy code

SELECT D.dnum AS deptno, COUNT(E.ssn) AS num\_employees, AVG(E.salary) AS avg\_salary

FROM DEPT D

JOIN EMPLOYEE E ON D.mgrssn = E.ssn

GROUP BY D.dnum;

c) Create a view that has pname, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

sql

Copy code

CREATE VIEW ProjectSummary AS

SELECT P.pname, D.dname AS controlling\_dept\_name, COUNT(W.ssn) AS num\_employees, SUM(W.hours) AS total\_hours

FROM PROJECT P

JOIN DEPT D ON P.dnumber = D.dnum

JOIN WORKON W ON P.pno = W.pno

GROUP BY P.pno, P.pname, D.dname

HAVING COUNT(W.ssn) > 1;

d) Create a procedure on EMPLOYEE table to determine the employees who will get a promotion (an employee will get a promotion after working on 5 projects).

sql

Copy code

DELIMITER //

CREATE PROCEDURE DeterminePromotion()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE ssn\_val INT;

DECLARE proj\_count INT;

-- Declare cursor for fetching employees

DECLARE cur CURSOR FOR SELECT ssn FROM EMPLOYEE;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

-- Open the cursor

OPEN cur;

-- Loop through employees

emp\_loop: LOOP

-- Fetch employee SSN

FETCH cur INTO ssn\_val;

IF done THEN

LEAVE emp\_loop;

END IF;

-- Count the number of projects for the employee

SELECT COUNT(\*) INTO proj\_count

FROM WORKON

WHERE ssn = ssn\_val;

-- Check if the employee qualifies for promotion

IF proj\_count >= 5 THEN

-- Print or perform action for promotion

PRINT CONCAT('Employee with SSN ', ssn\_val, ' is eligible for promotion.');

END IF;

END LOOP;

-- Close the cursor

CLOSE cur;

END//

DELIMITER ;

-- Usage:

CALL DeterminePromotion();

To create a data report showing the information of all female employees working in the "Research" department:

sql

Copy code

SELECT E.fname, E.lname, E.ssn, E.sex, E.salary, E.joindate, E.superssn, E.dno

FROM EMPLOYEE E

JOIN DEPT D ON E.dno = D.dnum

WHERE E.sex = 'F' AND D.dname = 'Research';

**Internal PrNorth Maharashtra LIniversity, Jalgaon.ctical Exam nation Feb-2023**

# CA-LAB I (NEW) : Lab on DBMS Duration 3:00 Hrs. Max marks : 20

#  Instructions:

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  + 1. Queries and their output.
    2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in eaCh table).

EMPLOYEE ( fna.me, Mame, ssn, sex, salary, joindate,superssn, dno4)

DEPT ( dname, dnum, mgrssn, dlocation)

PROJECT ( pname, pno, plocation, dnumber) WORK\_ ON ( ssn, pno, hours) Integrity Constraints:

* The values of any attributes should not be null.
* The deptno should less than 4 digit
* Ssn (social\_security\_no of employee)  Mgrssn(manager\_social\_security\_no)
* Superssn(supervisor\_social\_security\_no)

Queries:

* 1. Find the ssn of all employees who work on pno 101, 102 or 103.
  2. Make a list of all pno for project that involve an employee whose last name is 'sonar' either as a worker or as a manager of the dept that control the project.
  3. Write a trigger on insert on WORK\_ON table such that if total work hours of employee in company is less than 20 hours then his salary is deducted.
  4. Write a cursor on PROJECT table to fetch the first row from the table & display the total number of rows present in the table.

Create a data report showing the informatio.n of all the projects and names of employees working on individual projects.

CREATE TABLE EMPLOYEE (

fname VARCHAR(100) NOT NULL,

lname VARCHAR(100) NOT NULL,

ssn INT PRIMARY KEY,

sex CHAR(1) NOT NULL,

salary DECIMAL(10, 2) NOT NULL,

joindate DATE NOT NULL,

superssn INT,

dno INT NOT NULL,

CONSTRAINT CHK\_SSN CHECK (ssn < 10000),

CONSTRAINT CHK\_Sex CHECK (sex IN ('M', 'F')),

CONSTRAINT FK\_Employee\_SuperSsn FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn),

CONSTRAINT FK\_Employee\_Dept FOREIGN KEY (dno) REFERENCES DEPT(dnum)

);

CREATE TABLE DEPT (

dname VARCHAR(100) NOT NULL,

dnum INT PRIMARY KEY,

mgrssn INT NOT NULL,

dlocation VARCHAR(100) NOT NULL,

CONSTRAINT CHK\_DeptNo CHECK (dnum < 10000),

CONSTRAINT FK\_Dept\_Employee FOREIGN KEY (mgrssn) REFERENCES EMPLOYEE(ssn)

);

CREATE TABLE PROJECT (

pname VARCHAR(100) NOT NULL,

pno INT PRIMARY KEY,

plocation VARCHAR(100) NOT NULL,

dnumber INT NOT NULL,

CONSTRAINT CHK\_ProjectNo CHECK (pno < 10000),

CONSTRAINT FK\_Project\_Dept FOREIGN KEY (dnumber) REFERENCES DEPT(dnum)

);

CREATE TABLE WORK\_ON (

ssn INT NOT NULL,

pno INT NOT NULL,

hours INT NOT NULL,

CONSTRAINT FK\_WorkOn\_Employee FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn),

CONSTRAINT FK\_WorkOn\_Project FOREIGN KEY (pno) REFERENCES PROJECT(pno)

);

-- Integrity Constraints

ALTER TABLE EMPLOYEE ADD CONSTRAINT CHK\_Employee\_Salary CHECK (salary >= 0);

-- Queries

a) Find the ssn of all employees who work on pno 101, 102, or 103.

```sql

SELECT ssn

FROM WORK\_ON

WHERE pno IN (101, 102, 103);

b) Make a list of all pno for projects that involve an employee whose last name is 'sonar' either as a worker or as a manager of the department that controls the project.

sql

Copy code

SELECT DISTINCT P.pno

FROM PROJECT P

JOIN WORK\_ON W ON P.pno = W.pno

JOIN EMPLOYEE E ON W.ssn = E.ssn OR E.ssn = P.mgrssn

WHERE E.lname = 'sonar';

c) Write a trigger on insert on WORK\_ON table such that if the total work hours of an employee in the company is less than 20 hours, then their salary is deducted.

sql

Copy code

CREATE TRIGGER trg\_WorkOn\_Insert

AFTER INSERT ON WORK\_ON

FOR EACH ROW

BEGIN

DECLARE total\_hours INT;

-- Calculate total work hours for the employee

SELECT SUM(hours) INTO total\_hours

FROM WORK\_ON

WHERE ssn = NEW.ssn;

-- Check if total work hours is less than 20

IF total\_hours < 20 THEN

-- Deduct salary

UPDATE EMPLOYEE SET salary = salary - (NEW.hours \* 10) WHERE ssn = NEW.ssn;

END IF;

END;

d) Write a cursor on PROJECT table to fetch the first row from the table and display the total number of rows present in the table.

sql

Copy code

DECLARE @first\_row VARCHAR(100);

DECLARE @total\_rows INT;

DECLARE cur CURSOR FOR SELECT TOP 1 pname FROM PROJECT;

OPEN cur;

FETCH NEXT FROM cur INTO @first\_row;

CLOSE cur;

SELECT @total\_rows = COUNT(\*) FROM PROJECT;

PRINT 'First row: ' + @first\_row;

PRINT 'Total rows: ' + CONVERT(VARCHAR(10), @total\_rows);

To create a data report showing the information of all the projects and names of employees working on individual projects:

sql

Copy code

SELECT P.pname, E.fname, E.lname

FROM PROJECT P

JOIN WORK\_ON W ON P.pno = W.pno

JOIN EMPLOYEE E ON W.ssn = E.ssn;

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

CA-LAB I (NEW) : Lab on DBMS

Duration 3:00 Hrs. Max marks : 20

Instructions:

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

BOOKMASTER ( bid, title, author, price)

STUDENTMASTER (stud enrollno, sname,class,dept)

ACCESSIONTABLE (bid, accession\_no,avail)

ISSUETABLE(issueid,accession no,stud enrollno,issuedate,cluedate, ret\_date,bid)

Integrity Constraints:

* The values of any attributes should not be null.
* Avail should be T ( if book is not issue ) or F (if book is issue)

Queries:

* 1. Find the name of books which is issued maximum times.
  2. Find the detail information of books that are issued by computer department students.
  3. Write a procedure to calculate the fines for the books which are not return on or before due date. no.of days = (ret\_date - due\_date) fine = no.of days (ret\_date - due\_date) \* 10
  4. Write a trigger on insert of ISSUETABLE such that duedate = issuedate + 7

Create a data report that display the information of all books available in the library.

CREATE TABLE BOOKMASTER (

bid INT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

author VARCHAR(100) NOT NULL,

price DECIMAL(10, 2) NOT NULL

);

CREATE TABLE STUDENTMASTER (

enrollno INT PRIMARY KEY,

sname VARCHAR(100) NOT NULL,

class VARCHAR(100) NOT NULL,

dept VARCHAR(100) NOT NULL

);

CREATE TABLE ACCESSIONTABLE (

bid INT NOT NULL,

accession\_no INT NOT NULL,

avail CHAR(1) NOT NULL,

PRIMARY KEY (bid, accession\_no),

CONSTRAINT CHK\_Avail CHECK (avail IN ('T', 'F')),

CONSTRAINT FK\_Accession\_BookMaster FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)

);

CREATE TABLE ISSUETABLE (

issueid INT PRIMARY KEY,

accession\_no INT NOT NULL,

enrollno INT NOT NULL,

issuedate DATE NOT NULL,

cluedate DATE NOT NULL,

ret\_date DATE NOT NULL,

bid INT NOT NULL,

CONSTRAINT FK\_IssueTable\_Accession FOREIGN KEY (accession\_no) REFERENCES ACCESSIONTABLE(accession\_no),

CONSTRAINT FK\_IssueTable\_StudentMaster FOREIGN KEY (enrollno) REFERENCES STUDENTMASTER(enrollno),

CONSTRAINT FK\_IssueTable\_BookMaster FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)

);

-- Integrity Constraints

ALTER TABLE BOOKMASTER ADD CONSTRAINT CHK\_BookMaster\_Price CHECK (price >= 0);

-- Queries

a) Find the name of books which are issued maximum times.

```sql

SELECT B.title

FROM BOOKMASTER B

JOIN ACCESSIONTABLE A ON B.bid = A.bid

JOIN ISSUETABLE I ON A.accession\_no = I.accession\_no

GROUP BY B.bid, B.title

ORDER BY COUNT(\*) DESC

LIMIT 1;

b) Find the detailed information of books that are issued by computer department students.

sql

Copy code

SELECT B.title, B.author, B.price, S.sname, S.class, S.dept

FROM BOOKMASTER B

JOIN ACCESSIONTABLE A ON B.bid = A.bid

JOIN ISSUETABLE I ON A.accession\_no = I.accession\_no

JOIN STUDENTMASTER S ON I.enrollno = S.enrollno

WHERE S.dept = 'Computer';

c) Write a procedure to calculate the fines for the books that are not returned on or before the due date.

sql

Copy code

DELIMITER //

CREATE PROCEDURE CalculateFines()

BEGIN

DECLARE fine\_amount DECIMAL(10, 2);

DECLARE no\_of\_days INT;

SELECT DATEDIFF(NOW(), ret\_date) INTO no\_of\_days

FROM ISSUETABLE

WHERE ret\_date < DUE\_DATE;

SET fine\_amount = no\_of\_days \* 10;

-- Print or perform action with the fine amount

PRINT CONCAT('Fine amount: ', fine\_amount);

END//

DELIMITER ;

-- Usage:

CALL CalculateFines();

d) Write a trigger on the insert of ISSUETABLE such that duedate = issuedate + 7.

sql

Copy code

CREATE TRIGGER trg\_IssueTable\_Insert

BEFORE INSERT ON ISSUETABLE

FOR EACH ROW

BEGIN

SET NEW.cluedate = NEW.issuedate + INTERVAL 7 DAY;

END;

To create a data report that displays the information of all books available in the library:

sql

Copy code

SELECT B.bid, B.title, B.author, B.price, A.avail

FROM BOOKMASTER B

JOIN ACCESSIONTABLE A ON B.bid = A.bid;

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the  recOrds'aPPropriateliss6iliat-Menti6ned qUerieS can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form., trigger, assertion,cursor ,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

BOOKMASTER ( bid, title, author, price)

STUDENTMASTER (stud enrollno, snaine,class,dept)

ACCESSIONTABLE ( accession no,avail)

fiMUETABLE(issueid,as\_ussion no,stud earalno,issuedate,duedate, ret\_date,bi,d)

Integrity Constraints:

* The values of any attributes should not be null.
* Avail should be T ( if book is not issue ) or 1' (if book is issue)

Queries:

* 1. Find the detail information of the students who have issued books Between two given dates.
  2. Create a view that display all the accession infortnatiOn for a book having bid = 100
  3. Write a cursor to fetch last record from view **in (b).**
  4. Find the information of books issued by MCA students.

Create a input form for new book issue. Apply all possible validations.

-- Create tables

CREATE TABLE BOOKMASTER (

bid INT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

author VARCHAR(100) NOT NULL,

price DECIMAL(10, 2) NOT NULL

);

CREATE TABLE STUDENTMASTER (

enrollno INT PRIMARY KEY,

sname VARCHAR(100) NOT NULL,

class VARCHAR(100) NOT NULL,

dept VARCHAR(100) NOT NULL

);

CREATE TABLE ACCESSIONTABLE (

accession\_no INT PRIMARY KEY,

avail CHAR(1) NOT NULL,

CONSTRAINT CHK\_Avail CHECK (avail IN ('T', 'F'))

);

CREATE TABLE ISSUETABLE (

issueid INT PRIMARY KEY,

accession\_no INT NOT NULL,

enrollno INT NOT NULL,

issuedate DATE NOT NULL,

duedate DATE NOT NULL,

ret\_date DATE NOT NULL,

bid INT NOT NULL,

CONSTRAINT FK\_IssueTable\_Accession FOREIGN KEY (accession\_no) REFERENCES ACCESSIONTABLE(accession\_no),

CONSTRAINT FK\_IssueTable\_StudentMaster FOREIGN KEY (enrollno) REFERENCES STUDENTMASTER(enrollno),

CONSTRAINT FK\_IssueTable\_BookMaster FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)

);

-- Integrity Constraints

ALTER TABLE BOOKMASTER ADD CONSTRAINT CHK\_BookMaster\_Price CHECK (price >= 0);

-- Queries

a) Find the detailed information of the students who have issued books between two given dates.

```sql

SELECT S.enrollno, S.sname, S.class, S.dept, I.issuedate, I.duedate, I.ret\_date, B.title, B.author, B.price

FROM STUDENTMASTER S

JOIN ISSUETABLE I ON S.enrollno = I.enrollno

JOIN BOOKMASTER B ON I.bid = B.bid

WHERE I.issuedate BETWEEN 'start\_date' AND 'end\_date';

Replace 'start\_date' and 'end\_date' with the actual dates you want to query.

b) Create a view that displays all the accession information for a book having bid = 100.

sql

Copy code

CREATE VIEW AccessionInfo AS

SELECT A.accession\_no, A.avail

FROM ACCESSIONTABLE A

WHERE A.bid = 100;

c) Write a cursor to fetch the last record from the view in (b).

sql

Copy code

DECLARE @accession\_no INT;

DECLARE @avail CHAR(1);

DECLARE cur CURSOR FOR SELECT TOP 1 accession\_no, avail FROM AccessionInfo ORDER BY accession\_no DESC;

OPEN cur;

FETCH NEXT FROM cur INTO @accession\_no, @avail;

CLOSE cur;

-- Print or perform action with @accession\_no and @avail values

PRINT CONCAT('Accession No:', @accession\_no, ', Avail:', @avail);

d) Find the information of books issued by MCA students.

sql

Copy code

SELECT B.bid, B.title, B.author, B.price, S.enrollno, S.sname, S.class, S.dept

FROM BOOKMASTER B

JOIN ISSUETABLE I ON B.bid = I.bid

JOIN STUDENTMASTER S ON I.enrollno = S.enrollno

WHERE S.dept = 'MCA';

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

# Duration 3:00 Hrs. Max marks : 20

# Instructions:

* **Read** the slip carefully.
* Read the Schemas carefully before tilling records in the table & fill the  records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the followingqueries using SQL. (Fill up database with at least 10 records ir+ eaoh,tab4e).

BOOKMASTER ( bid, title, author, price)

STUDENTMASTER (stud enroll no, sname,class,dept)

ACCESSIONTABLE ( bid, accession no,avail)

SUETABLE(issueid,accessipn no.stl.KLmrsillag,issuedate,cluedate, ret\_date,hisp

Integrity Constraints:

* The values of any attributes should not be null.
* Avail should be T ( if book is not issue ) or F (if book i8 issue)

Queries:

1. Write a procedure for giving the detail information of books available in the library.
2. Find the number of books issued by each student.
3. Write a trigger such that the return date should not exceed today's date.
4. Find the number of books available in the library & written by "Henry Korth".

Create a class wise issue report of books.

CREATE TABLE BOOKMASTER (

bid INT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

author VARCHAR(100) NOT NULL,

price DECIMAL(10, 2) NOT NULL

);

CREATE TABLE STUDENTMASTER (

enrollno INT PRIMARY KEY,

sname VARCHAR(100) NOT NULL,

class VARCHAR(100) NOT NULL,

dept VARCHAR(100) NOT NULL

);

CREATE TABLE ACCESSIONTABLE (

bid INT NOT NULL,

accession\_no INT PRIMARY KEY,

avail CHAR(1) NOT NULL,

CONSTRAINT CHK\_Avail CHECK (avail IN ('T', 'F')),

CONSTRAINT FK\_Accession\_BookMaster FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)

);

CREATE TABLE ISSUETABLE (

issueid INT PRIMARY KEY,

accession\_no INT NOT NULL,

enrollno INT NOT NULL,

issuedate DATE NOT NULL,

cluedate DATE NOT NULL,

ret\_date DATE NOT NULL,

CONSTRAINT FK\_IssueTable\_Accession FOREIGN KEY (accession\_no) REFERENCES ACCESSIONTABLE(accession\_no),

CONSTRAINT FK\_IssueTable\_StudentMaster FOREIGN KEY (enrollno) REFERENCES STUDENTMASTER(enrollno)

);

-- Integrity Constraints

ALTER TABLE BOOKMASTER ADD CONSTRAINT CHK\_BookMaster\_Price CHECK (price >= 0);

-- Queries

a) Write a procedure for giving the detailed information of books available in the library.

```sql

CREATE PROCEDURE GetAvailableBooks()

BEGIN

SELECT B.bid, B.title, B.author, B.price

FROM BOOKMASTER B

JOIN ACCESSIONTABLE A ON B.bid = A.bid

WHERE A.avail = 'T';

END;

b) Find the number of books issued by each student.

sql

Copy code

SELECT S.enrollno, S.sname, COUNT(\*) AS num\_books\_issued

FROM STUDENTMASTER S

JOIN ISSUETABLE I ON S.enrollno = I.enrollno

GROUP BY S.enrollno, S.sname;

c) Write a trigger such that the return date should not exceed today's date.

sql

Copy code

CREATE TRIGGER trg\_IssueTable\_ReturnDate

BEFORE INSERT ON ISSUETABLE

FOR EACH ROW

BEGIN

IF NEW.ret\_date > CURDATE() THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Return date should not exceed today''s date.';

END IF;

END;

d) Find the number of books available in the library and written by "Henry Korth".

sql

Copy code

SELECT COUNT(\*) AS num\_books\_available

FROM BOOKMASTER B

JOIN ACCESSIONTABLE A ON B.bid = A.bid

WHERE A.avail = 'T'

AND B.author = 'Henry Korth';

To create a class-wise issue report of books, you can use the following query:

sql

Copy code

SELECT S.class, B.title, B.author, I.issuedate, I.ret\_date

FROM STUDENTMASTER S

JOIN ISSUETABLE I ON S.enrollno = I.enrollno

JOIN ACCESSIONTABLE A ON I.accession\_no = A.accession\_no

JOIN BOOKMASTER B ON A.bid = B.bid;

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks : 20**

# Instructions:

* **Read** the slip carefully.
* Read the Schemas carefully before tilling records in the table & fill the  records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  + 1. Queries and their output.
    2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in •aa table).

EMPLOYEE ( fname, lname, ssn, sex, salary, joindate,superssn, dno,)

DEPT ( dname, dnum, mgrssn, dlocation)

PROJECT ( pname, pno, plocation, &lumber) WORK ON ( ssn, pno, hours)

Integrity Constraints:

* The values of any attributes should not be null.
* The deptno should less than 4 digit
* ssn (social security\_no of employee)
* mgrssn(manager\_social\_security\_no)
* superssn(supervisor\_social\_security\_no)

Queries:

* 1. For every project located in `jalgaon'. List the pno,the controlling detptno and dept manager last name.
  2. For each project on which more than two employee work, Find the pno, pnarne & no. of employees who work on the project.
  3. create a view that has the deptna.me,manager name *8z* manager salary for every dept.
  4. Express the following constraint as SQL assertions -

"salary of employee must not be greater than the salary of the manager of the dept".

Create a data entry for New employee entry. Apply all possible validations.

**4. -- Create tables**

**CREATE TABLE EMPLOYEE (**

**fname VARCHAR(100) NOT NULL,**

**lname VARCHAR(100) NOT NULL,**

**ssn INT PRIMARY KEY,**

**sex CHAR(1) NOT NULL,**

**salary DECIMAL(10, 2) NOT NULL,**

**joindate DATE NOT NULL,**

**superssn INT,**

**dno INT,**

**CONSTRAINT FK\_Employee\_SuperSsn FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn),**

**CONSTRAINT CHK\_Employee\_Sex CHECK (sex IN ('M', 'F'))**

**);**

**CREATE TABLE DEPT (**

**dname VARCHAR(100) NOT NULL,**

**dnum INT PRIMARY KEY,**

**mgrssn INT,**

**dlocation VARCHAR(100) NOT NULL,**

**CONSTRAINT FK\_Dept\_ManagerSsn FOREIGN KEY (mgrssn) REFERENCES EMPLOYEE(ssn)**

**);**

**CREATE TABLE PROJECT (**

**pname VARCHAR(100) NOT NULL,**

**pno INT PRIMARY KEY,**

**plocation VARCHAR(100) NOT NULL,**

**dnumber INT,**

**CONSTRAINT FK\_Project\_Dept FOREIGN KEY (dnumber) REFERENCES DEPT(dnum)**

**);**

**CREATE TABLE WORK\_ON (**

**ssn INT NOT NULL,**

**pno INT NOT NULL,**

**hours INT NOT NULL,**

**CONSTRAINT FK\_WorkOn\_Employee FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn),**

**CONSTRAINT FK\_WorkOn\_Project FOREIGN KEY (pno) REFERENCES PROJECT(pno),**

**CONSTRAINT PK\_WorkOn PRIMARY KEY (ssn, pno)**

**);**

**-- Integrity Constraints**

**ALTER TABLE EMPLOYEE ADD CONSTRAINT CHK\_Employee\_Salary CHECK (salary >= 0);**

**-- Queries**

**a) For every project located in 'jalgaon', list the pno, the controlling deptno, and the dept manager's last name.**

**```sql**

**SELECT P.pno, P.dnumber, E.lname AS manager\_last\_name**

**FROM PROJECT P**

**JOIN DEPT D ON P.dnumber = D.dnum**

**JOIN EMPLOYEE E ON D.mgrssn = E.ssn**

**WHERE P.plocation = 'jalgaon';**

**b) For each project on which more than two employees work, find the pno, pname, and the number of employees who work on the project.**

**sql**

**Copy code**

**SELECT P.pno, P.pname, COUNT(\*) AS num\_employees**

**FROM PROJECT P**

**JOIN WORK\_ON W ON P.pno = W.pno**

**GROUP BY P.pno, P.pname**

**HAVING COUNT(\*) > 2;**

**c) Create a view that has the deptname, manager name, and manager salary for every dept.**

**sql**

**Copy code**

**CREATE VIEW DeptManagerView AS**

**SELECT D.dname, E.fname, E.lname, E.salary**

**FROM DEPT D**

**JOIN EMPLOYEE E ON D.mgrssn = E.ssn;**

**d) Express the following constraint as SQL assertions - "salary of employee must not be greater than the salary of the manager of the dept".**

**sql**

**Copy code**

**CREATE ASSERTION SalaryConstraint CHECK (**

**NOT EXISTS (**

**SELECT 1**

**FROM EMPLOYEE E**

**JOIN DEPT D ON E.ssn = D.mgrssn**

**WHERE E.salary > D.salary**

**)**

**);**

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

# CA-LAB I (NEW) : Lab on DBMS Duration 3:00 Hrs. Max marks : 20

# Instructions:

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be .executed.
* The underline attributes is primary key &.double underline attributes are foreign keys.
* Take Printout of:
  + 1. Queries and their output.
    2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Creat4 database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

ACCOUNT (accno, open\_date, acctype, balance)

TRANSACTION (trans id, trans date, accno. trans\_type, amount) CUSTOMER

(cust id name, address, accno)

Integrity Constraints:

* The values of any attributes should not be null.
* acctype value should be P(Personal) or J(Joint).
* Accno should be less than 3 digits.
* Trans\_type should be C(Credit) or D(Debit)

Queries:

* 1. Find the details of customers who have opened the accounts within the period 25-32006 to 28-3-2006.
  2. Find the details of customers who have joint accounts & balance is less than 2 lakhs.
  3. Write a trigger TRANSACTION on table to calculate the current balance of the account on which transaction is made.

( if trans\_type = c then bal = bal + amt else if trans\_type = d then bal bal — amt)

* 1. write a cursor on CUSTOMER table to fetch the last row.

Create a data entry for New customer entry. Apply all possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks: 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* 'fake Printout of:
  + - 1. Queries and their output.
      2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

PRODUCT ( Maker, .Modelno, Type )

PC( 4.oc jeki,o, Speed, RAM, HD, CD, Price )

LAPTOP ( Modelno, Speed, RAM, HD, Price )

PRINTER ( Mosklm, Color, Type, Price )

Details regarding Schemas

* PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.
* The value for Maker in Product table can be IBM, Compaq,etc.
* PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

* The values of any attributes should not be null.
* Product Type should one of these (PC, Laptop or Printer) Queries:

1. Find the manufacturers of color printers.
2. Find the laptops whose speed is slower than that of any PC.
3. Express the following constraint as SQL assertions -

"No black & white printer should have price greater than color printers."

1. write a trigger on PC & LAPTOP table such that the hard disk size should be greater than 20 GB

Design an input form for entering PRINTER data. Apply possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 firs. Max marks: 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:

I) Queries and their output.

2) Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

**H . S** V **1 • 1 , 1 1 4 t a i t** • **I . 1 1 . 1 n i ) 0 I I ( V I ' e h**

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

PRODUCT ( Maker, Modelno, Type )

PC ( IvIodeln, Speed, RAM, HD, CD, Price )

LAPTOP (Rh, Speed, RAM, HD, Price ) PRINTER ( Model, Color, Type, Price )

Details regarding Schemas

* PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.
* The value for Maker in Product table can be IBM, Compaq,etc.
* PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

* The values of any attributes should not be null.
* Product Type should one of these (PC, Laptop or Printer) Queries:

1. Find the manufacturers of color printers.
2. Find the laptops whose speed is slower than that of any PC.

C) Express the following constraint as SQL assertions -

No black & white printer should have price greater than color printers."

d) write a trigger on PC & LAPTOP table such that the hard disk size should be greater than 20 GB

Design an input form for entering PC data. Apply possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

# CA-LAB I (NEW) : Lab on DBMS Duration 3:00 Hrs. Max marks: 20

* \_
* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:

1. Queries and their output.
2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

PRODUCT ( Maker, Modelno, Type )

PC ( Modelno, Speed, RAM, HD, CD, Price )

LAPTOP ( Modelno, Speed, RAM, HD, Price )

PRINTER ( Modelno, Color, Type, Price )

Details regarding Schemas

* PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.
* The value for Maker in Product table can be IBM, Compaq,etc.
* PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

* The values of any attributes should not be null.
* Product Type should one of these (PC, Laptop or Printer).

Queries:

1. Find the different types of printers produced by Epson.
2. Find those hard disk sizes which occur in two or more PC's.
3. Write a trigger on LAPTOP table such that the minimum speed should be 120MHz.
4. Demonstrate the use of cursor using PRODUCT table.

Design an input form for entering LAPTOP data. Apply possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks: 20**

* + - * Read the slip carefully.
      * Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.

The underline attributes is primary key & double underline attributes are foreign keys.

* + - * Take Printout of:
    1. Queries and their output.
    2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

\*PRODUCT ( Maker, Modelno, Type )

PC ( Modelno, Speed, RAM, HD, CD, Price )

LAPTOP ( Modelno, Speed, RAM, HD, Price )

PRINTER (1Viodelno, Color, Type, Price )

Details regarding Schemas

* + PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.
  + The value for Maker in Product table can be IBM, Compacketc.
  + PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

* + The values of any attributes should not be null.
  + Product Type should one of these (PC, Laptop or Printer)

Queries:

1. Find the different types of printers produced by Epson.
2. Find those hard disk sizes which occur in two or more PC's.
3. Write a trigger on LAPTOP table such that the minimum speed should be 1201v1Hz.
4. Demonstrate the use of cursor using PRODUCT table.

Design an input form for entering LAPTOP data. Apply possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

# Duration 3:00 Hrs. Max marks: 20

# Instructions:

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key &. double underline attributes are foreign keys.
* Take Printout of:

I) Queries and their output.

2) Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

PRODUCT( Maker, Modelno, Type )

PC ( Modelno, Speed, RAM, HD, CD, Price )

LAPTOP ( Modetno, Speed, RAM, HD, Price ) PRINTER ( ModelaQ, Color, Type. Price )

Details regarding Schemas

* PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.
* The value for Maker in Product table can be 113M, Compaq,etc.
* PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

* The values of any attributes should not be null.
* Product Type should one of these (PC, Laptop or Printer)

Queries:

* 1. Find PC models having a speed of at least 150 MHz.
  2. Find those manufacturers that sell Laptops, but not PC's.
  3. Write a trigger on LAPTOP table such that the price should not less than 30000
  4. Write a procedure to find the manufacturer who has produced the most expensive laptop.

Design an input form for entering LAPTOP data. Apply possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS Duration 3:00 firs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key & double underline attributes are foreign keys.
* Take Printout of:
  1. Queries and their output.
  2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification)

PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did)

ADMITTEDPATIENT (Pcode, Entry date, Discharge\_clate, wardno, disease )

Integrity Constraints:

* The values of any attributes should not be null.
* Gender value should be M (male) or F(feinale).
* Wardno should be less than 6.

Queries:

* 1. Find the details of patient who are admitted within the period 03/03/08 to 25/ 03/08.
  2. Find the names of doctors who are treating Jalf\_zaon patients.
  3. write a procedure on ADMITTEDPATIENT table such as to calculate the bill of all patients currently admitted in the hospital.

(bill = no \_ of\_ days \* 500)

* 1. Write a trigger on Doctor table such that the specialization should be :- M.B.B.S./B.A.M.S/M.S.

Create a data entry form for new DOCTOR. Apply all possible validations.

**RCPET’s IMRD, Shirpur**

**Internal Practical Examination Feb-2023**

**CA-LAB I (NEW) : Lab on DBMS**

**Duration 3:00 Hrs. Max marks : 20**

**Instructions:**

* Read the slip carefully.
* Read the Schemas carefully before filling records in the table & fill the records appropriately so that mentioned queries can be executed.
* The underline attributes is primary key &, double underline attributes are foreign keys.
* Take Printout of:
  + 1. Queries and their output.
    2. Code if written by you for generating report, input form, trigger, assertion, cursor,views along with their output.

Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification)

PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did)

ADMITTEDPATIENT (Pcode, Entry date, Discharge\_date, warcino, disease )

Integrity Constraints:

* The values of any attributes should not be null.
* Gender value should be .M (male) or F(female).
* Wardno should be less than 6. Queries:
  + 1. Find details of the patients who are treated by M.B.B.S. doctors.
    2. Find name of the doctor who are treating the male patients suffering from disease brain tumor & having age less than 40 years.
    3. write a procedure on ADMITTEDPATIENT table such as to calculate the bill of all patients who are discharged on 30-3-2008 .

(bill no of days \* 500)

* + 1. write a cursor on DOCTOR table to fetch the first row & display the number of rows present in the table.

Create a data entry form for new patient. Apply all possible validations.