CustomerID INT PRIMARY SET 1 SFT 2 4) <?php } else { echo "No data SET 3 GRANT SELECT ON } catch (Exception e) { SET 4 (1002 'Prof Williams' found<br>"; employee TO user1 WITH (ClassNotFoundException e) 'Electrical Engineering', KEY, 1)CREATE TABLE 1)I)alter by adding column 2) i)select // Replace these variables echo "Error creating 2) i) SELECT \* e printStackTrace(): GRANT OPTION: true) with your own MySQL name,gender,mobilenc STUDENT\_DETAILS CustomerName FROM students GRANT SELECT ON CREATE TABLE EMPLOYEE ( alter table employe fromm students order by database credentials "<br>"; e.printStackTrace() VARCHAR(255), // Close connection RegisterNumber INT WHERE score > (SELECT employee TO user1 WITH add column salary int(10); Email VARCHAR(255 GRANT OPTION; \$conn->close(); MAX(score) FROM "vour servername" KEY. ii)alter table by modifying students order by age desc // Insert data into the table students); StudentNami REVOKE SELECT ON column name: VARCHAR(255), \$insertDataQuery = "INSERT ii) SELECT student id subject,count(regno) fron employee FROM user1; INSERT INTO CUSTOMER alter table employee "your username"; VARCHAR(255), INTO example table (name Denartmen (CustomerID rename column name to students group by subject: FROM assessments 4) import // JDBC variables for age) VALUES ('John Doe', VARCHAR(50), Position VARCHAR(50), \$password = CustomerName, Email java.sql.Connection; employees; 3)CREATE TABLE Class ( opening, closing, and "vour password": GROUP BY student id VALUES BatchYear INT Salary DECIMAL(10, 2) iii)desc employee import StudentID INT PRIMARY if (\$conn-HAVING COUNT(\*) > 1; (4001, 'Eva Green java.sql.DriverManager; rename table employee to try (Connection >query(\$insertDataQuery) === TRUE) { "your database"; 'eva@example.com'), iii) SELECT faculty\_name, connection = DriverManager.getConnect import java.sql.Statement INSERT INTO INSERT INTO EMPLOYEE delete from emp where (4002, 'Michael Johnson ioin date STUDENT DETAILS (EmployeeID. VARCHAR(255), no=2; echo "Data inserted on(JDBC\_URL, USERNAME. 'michael@example.com'): \$conn = new mysqli(\$servername, (RegisterNumber, StudentName, Department, EmployeeName, Position Salary) VALUES drop table emp successfully<br> Grade CHAR(1 UPDATE STUDENT\_DETAILS WHERE join date = (SELECT 2)select bank from Susername, Spassword. } else { MvSQLIDBCExample { Statement BatchYear) VALUES MAX(join date) FROM echo "Error insertina faculty): (191711342, 'John Doe' 'Manager', 60000.00) 'Updated Name' INSERT INTO Class connection.createStatemer 3)select customername // Check connection data: " Sconn-serror 'Computer Science', 2019), (StudentID\_StudentName // JDBC URL. username t()) { (2002 'Rob Johnson WHERE RegisterNumber = from customers Grade) VALUES (191711343, 'Jane Smith', if (\$conn->connect error) { and password of MySQL 'Software Engineer', 191711342; 3)CREATE TABLE employee ( 4)select \* from bank where 'Electrical Engineering', 50000.00). (1. 'Alice'. 'A'). die("Connection failed: " DELETE FROM FACULTY // SQL query to employee id SERIAL // Select data from the CREATE TABLE BANK Sconn->connect error): private static final String (2, 'Bob', 'B'). create the table WHERE ResignedJob = true JDBC\_URL = "jdbc:mysql://localhost:330 CREATE TABLE FACULTY ( balance>10000; (3, 'Charlie', 'C'): String SELECT \* FROM select \* from bank where first\_name VARCHAR(50). \$selectDataQuery = "SELECT EmployeeID INT PRIMARY PRIMARY KEY, // Create a table 6/examDB"; createTableQuery = STUDENT\_DETAILS; balance between 10000 and SAVEDOINT Savennint1 id, name, age FROM example\_table"; KEY last name VARCHAR(50), "CREATE TABLE JavaLah private static final String ScreateTableQuery = SELECT \* FROM FACULTY: INSERT INTO Class department (Name TEXT(16), Surname FacultyName VARCHAR(255). "CREATE TABLE IF NOT USERNAME = (StudentID, StudentName Grade) VALUES \$result = \$conn->query(\$selectDataQuery) VARCHAR(50), TEXT(6), City TEXT(6))"; VARCHAR(255), EXISTS example\_table "your\_username" salary DECIMAL(10, 2) Department 2 CREATE TABLE STUDENT ( Id INT AUTO INCREMENT private static final String if (\$result->num\_rows > 0) { VARCHAR(50), (4, 'David', 'A') PASSWORD = // Execute the query PRIMARY KEY, StudentID INT PRIMARY INSERT INTO BANK (5, 'Eva', 'B'); echo "Selected "your\_password" ResignedJob BOOLEAN KEY, name VARCHAR(255) INSERT INTO employee (AccountNumber data:<br/>br>": statement.executeUpdate(o SAVEPOINT Savepoint2; NOT NULL, CustomerName, Balance) (first name, last name while (Śrow = Śresult reateTableQuery); VALUES VARCHAR(255). department, salary) VALUES INSERT INTO Class age INT public static void INSERT INTO FACULTY (3001, 'Mary Davis', (StudentID, StudentName main(String[] args) { ('John', 'Doe', 'IT' (EmployeeID, FacultyNam )": echo "ID: " Srow["id"] Grade) VALUES 60000.00), Department, ResignedJob) 1500.00) // Load the MySOL CREATE TABLE FACULTY ( if (\$conn-(6, 'Frank', 'C'), System.out.println("Table (3002, 'Samuel Wilson', ('Jane', 'Smith', 'HR' >query(ScreateTableQuery) ' - Age: " . \$row["age"]. FacultyID INT PRIMARY 'JavaLab' created (1001, 'Dr. Johnson' 3000.00). 55000.00), (7, 'Grace', 'A'); "chro" try { Class.forName("com.mysql successfully."); KEY, 'Computer Science', false), CREATE TABLE CUSTOMER ( ('Bob', 'Johnson ROLLBACK TO Savenoint 2: echo "Table created FacultyName ci.idbc.Driver"): 'Finance', 70000.00); VARCHAR(255), SELECT \* FROM Class; } else {

DepartmentID INT FORFIGN KEY REFERENCES DEPARTMENT/D D) CREATE TABLE DEPARTMENT ( DepartmentID INT PRIMARY KEY, DenartmentNam VARCHAR(50)

CREATE TABLE COURSE CourseNam

VARCHAR(255), FOREIGN KEY (FacultyID) FACULTY(FacultyID)

CREATE TABLE SCORES ( ScoreID INT PRIMARY StudentID INT.

CourseID INT, Score INT. FOREIGN KEY (StudentID) REFERENCES STUDENT(StudentID)

FOREIGN KEY (CourseID)
REFERENCES COURSE(CourseID)

SELECT DISTINCT d.DepartmentNam FROM FACULTY f INNER JOIN DEPARTMENT d ON f.DepartmentID = d.DepartmentID: SELECT & StudentName

INSERT INTO DEPARTMENT

DenartmentName) VALUES

(2, 'Electrical Engineering'),

(1, 'Computer Science'),

(DepartmentID,

(3. 'Mechanical

Engineering'),

(4, 'Finance');

INSERT INTO FACULTY

(FacultyID, FacultyNan

DepartmentID) VALUES

(102, 'Prof. Johnson', 2).

INSERT INTO STUDENT

(201, 'Alice Brown').

(203 'Charlie Smith'):

INSERT INTO COURSE

FacultyID) VALUES

(301, 'Introduction to

Programming', 101),

(303, 'Mechanics of

INSERT INTO SCORES

(401, 201, 301, 90),

(402, 202, 302, 85).

(403, 203, 303, 92);

for op⊆

(ScoreID, StudentID, CourseID, Score) VALUES

Materials', 103);

(302 'Circuit Analysis' 102)

(CourseID. CourseName

(202, 'Bob Davis'),

(StudentID, StudentName)

(101, 'Dr. Smith', 1).

(103, 'Dr. Lee', 3);

VALUES

c.CourseNumber FROM STUDENTS LEFT JOIN SCORES SC ON s.StudentID = sc.StudentII LEFT JOIN COURSE c ON WHERE sc.Score IS NULL: iii)

SELECT c CourseName FROM COURSE c c FacultyID = f FacultyID INNER JOIN DEPARTMENT d ON f.DepartmentID = WHERE d DenartmentName

= 'ECE' AND f.FacultyName = 'Kamal'; 3) I)DELIMITER // CREATE PROCEDURE

BEGIN

student info WHERE lid IN (100, 101, 105, 110)

ELSEIF creditLimit >: 50000 THEN SET p\_customerLevel = 'GOID' ELSE SET p\_cus 'SILVER' END IF END // DELIMITER

THEN

'PLATINUM':

(Subject = 'Computers' OR Subject = 'History' OF Subject = 'Literature'] DELIMITER

END //

OUT n customerleve

DECLARE creditLimit

VARCHAR(10))

DECIMAL(10, 2);

creditLimit FROM

customers WHERE

CustomerNumber =

p\_customerNumher

SET p\_customerLevel

DECLARE customerLevel VARCHAR(10): CALL GetStudentInfo(): CREATE PROCEDURE -- Set the input parameter GetCustomerLevel(II p customerNumber INT

SET customerNumber = 123; -- Replace with an actual customer number

-- Declare variables to

capture output parameters

DECLARE customerNumbe

-- Call the GetCustomerLeve procedure

GetCustomerLevel(custome @customerLevel); -- Retrieve the output

SELECT @customerLevel AS 4) CREATE TABLE Employee

EmployeeID INT PRIMARY KEY. Name VARCHAR(255). DateOfBirth DATE. DepartmentID INT PositionID INT.

- other attributes

CREATE TABLE Departmen

PRIMARY KEY, VARCHAR(255).

CREATE TABLE Position ( PositionID INT PRIMARY KEY,

PositionName VARCHAR(255). - other attributes

ALTER TABLE Employee ADD FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID) ALTER TABLE Employee

ADD FOREIGN KEY (PositionID) REFERENCES Position(PositionID); INSERT INTO Employee (EmployeeID, Name, DateOfBirth, DepartmentID, PositionID) VALUES (1, 'John Doe

'1990-01-01', 101, 201) (DepartmentID VALUES (101, 'Human

INSERT INTO Position (PositionID, PositionName VALUES (201, 'Manager'); SELECT \* FROM Employee

WHERE DepartmentID = 101: SELECT Employee.\*, Department.DepartmentNa me. Position.PositionName

INSERT INTO STUDENT FROM Employee (RegisterNumber, Name Employee DepartmentID = -> VALUES (1, 'John Doe', '1990-JOIN Position ON Employee.PositionID = Position.PositionID; -> (2. 'Jane Smith'.

SET 5

2) i) CREATE TABLE

-> RegisterNumber INT

-> Name VARCHAR(50)

STUDENT (

PRIMARY KEY,

-> DOB DATE

'1995-05-20'), -> (3, 'Bob Johnson' '1998-12-03'); SELECT \* FROM std\_view; ii) CREATE TABLE Persons ( -> PersonID INT

CREATE INDEX idx\_lastname

Persons(LastName)

-> FROM Persons

SELECT \*

PRIMARY KEY, -> LastName VARCHAR(50) 2) INSERT INTO Perso (PersonID, LastName) -> VALUES

-> (1, 'Doe'), VALUES (1. 'Customer1 -> (2, 'Smith'), 75000.00), -> (3 'lohnson')

30000.00). // 'Customer/

-> WHERE LastName = iii)CREATE TABLE YourTable

-> ID INT AUTO\_INCREMENT PRIMARY KEY.

-> Name VARCHAR(50) INSERT INTO YourTable (Name) VALUES ('John

INSERT INTO YourTable (Name) VALUES ('Jane Smith');

INSERT INTO YourTable (Name) VALUES ('Bob SELECT \* FROM YourTable;

3)CREATE TABLE cno INT PRIMARY KEY, cname VARCHAR(50), creditlimit DECIMAL(10

INSERT INTO CUSTOMER (cno, cname, creditlimit)

(3, 'Customer3', 8000.00)

45000.00),

(5, 'Customer5' Name VARCHAR(100), 120000.00); Location VARCHAR(100) YearFounded INT @@GLOBAL.max sp recurs ion\_depth = 255; DELIMITER // CREATE TABLE Books

OUT result BIGINT

SET result = 1:

ComputeFactorial(n - 1.

SET result = n \* result;

CALL ComputeFactorial(5,

SELECT @factorial\_result AS

4) CREATE TABLE Authors (

AuthorID INT PRIMARY

FirstName VARCHAR(50).

@factorial result):

factorial\_result;

BEGIN

ELSE

CALL

END IF:

DELIMITER :

FND //

CREATE PROCEDURE BookID INT PRIMARY KEY, ComputeFactorial(IN n INT, Title VARCHAR(100), ISBN VARCHAR(13) Genre VARCHAR(50), IF n = 0 OR n = 1 THEN

> PublisherID INT FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID), FOREIGN KEY (PublisherID) REFERENCES

Publishers(PublisherID) INSERT INTO Author (AuthorID, FirstName LastName, BirthDate)

VALUES (1, 'John', 'Doe', '1980-05-15'),

(2, 'Jane', 'Smith', '1975-11-227: INSERT INTO Publishers (PublisherID, Name, Location, YearFounded)

LastName VARCHAR(50). VALUES (1. 'ABC Publishers', 'New York', 1990), CREATE TABLE Publishers (

(2, 'XYZ Publishers', 'London', 1985); INSERT INTO Books (BookID Title, ISBN, Genre AuthorID, PublisherID) VALUES

(101, 'Introduction to '1234567890123' 'Education', 1, 1); TO GET OUTPUT SELECT Rooks Title Books.ISBN, Books.Genre,

Authors FirstName Authors.LastName Authors.BirthDate. Publishers.Name AS

Publisher Publishers.Location Publishers.YearFounded IOIN Authors ON Books.AuthorID = Authors.AuthorID

JOIN Publishers ON Books.PublisherID : Publishers.PublisherID

NewName VARCHAR(50), SET 6 SELECT SELECT \* FROM Hospital SET 7 -> (4, 'Alice', 'Williams' SELECT \* FROM SET 8 (102, 'Boat2', 'green'), VALUES ALTER TABLE CUSTOMER 35000); EMPLOYEES\_AUDIT 1) I JAITER TARIE WORKERS -> employee id 1) i)SELECT AVG(Marks) AS OldEmail VARCHAR(50) CREATE TABLE SALLORS ( (103 'Boat3' 'blue'): (1 'lohn' 8 25) MODIFY COLUMN Cust city -> \$\$ DELIMITER // ADD COLUMN Salary INT; 4) CREATE TABLE IF NOT NewEmail VARCHAR(50) sid INT PRIMARY KEY, INSERT INTO RESERVES (sid, -> first name, (2, 'Alice', 7, 22), EXISTS bill ( FROM STUDENT SELECT 4)CREATE TABLE DEPT ( bid. dav) iv)ALTER TABLE CUSTOMER CREATE PROCEDURE ii)ALTER TABLE WORKERS -> last\_name ChangeTimestam sname VARCHAR(255) (3, 'Bob', 7, 30) bill\_id INT WHERE Coursename : -> employee\_id, TIMESTAMP DEFAULT Deptno INT PRIMARY KEY VALUES DROP COLUMN salary; CHANGE COLUMN Name -> salary rating INT (4 'Eva' 9 21) AUTO INCREMENT 'C002'-CURRENT TIMESTAME REGIN -> first\_name (1, 101, '2023-01-01'), -> (CASE age INT (5, 'Charlie', 8, 28), VARCHAR(255); ii)SELECT Coursename DECLARE done INT -> WHEN salary > -> last name. Location VARCHAR(255) (2. 102. '2023-01-02'). 4)CREATE TABLE EMPLOYEE customer name MAY(Marks) AS MayMark (6, 'Diana', 9, 23); DEFAULT FALSE; DELIMITER // MIN(Marks) AS MinMarl 50000 THEN 'High Salary' -> salary, (3, 101, '2023-01-03'), DESCRIBE WORKERS CREATE TABLE BOATS ( SELECT rating, AVG(age) AS DECLARE email CREATE TRIGGER EmnNo INT PRIMARY KEY CREATE TABLE emp AS amount DECIMAL(10, 2), FROM STUDENT INSERT INTO DEPT (Deptno, AverageAge VARCHAR(50); -> (CASE (4, 103, '2023-01-04'); before\_update\_employees hid INT PRIMARY KEY SELECT \* FROM WORKERS; 30000 THEN 'Medium DName, Location) Name VARCHAR(50), payment date DATE WHERE Coursename : -> WHEN salary > FROM SAILORS BEFORE UPDATE ON bname VARCHAR(255), iv) DELETE FROM em 50000 THEN 'High Salary' VALUES EMPLOYEES 1) I)SELECT DISTINCT S.sid GROUP BY rating -> ELSE 'Low Salary' color VARCHAR(255) VARCHAR(50). GROUP BY Coursename: -> WHFN salary : (1, 'HR', 'New York'), DROP TABLE emp INSERT INTO bill FOR FACH ROW EMPLOYEE; 30000 THEN 'Mediun (customer\_name, amount iii)SELECT SUM(Marks) AS (2, 'IT', 'San Francisco'), salary category Salary' BEGIN JOIN RESERVES R ON S.sid = payment date) VALUES CREATE TABLE RESERVES Rank Incation (3, 'Finance', 'London'), 2)CREATE TABLE employees -> FROM -> ELSE 'Low Salary' INSERT INTO VARCHAR(50), 3)CREATE TABLE FROM STUDENT; ('Customer1', 100.00, sid INT EMPLOYEES AUDIT (4, 'Marketing', 'Paris'); JOIN BOATS B ON R.bid = CUSTOMER ( SET done = TRUE: '2023-01-01'), -> FND) AS (EmployeeID, OldNam bid INT, OPEN cur: -> employee\_id INT salary\_category I)RENAME TABLE DEPT TO Cust name ('Customer2', 150.50, -> \$\$ NewName, OldEmail. 2)CREATE TABLE employees DEPARTMENT WHERE B.color IN ('red', VARCHAR2(20) day DATE -> first name '2023-01-05'). -> FROM NewEmail) read loop: LOOP VARCHAR(50), ii)ALTER TABLE FOREIGN KEY (sid) Cust street VALUES (OLD.ID, ('Customer3', 200.75 -> employees; ii)SELECT sname, age 3)CREATE DATABASE IF NOT -> employee id INT, DEPARTMENT REFERENCES SAILORS(sid VARCHAR2(20), -> last name '2023-01-10') OLD Name NEW Name -> \$\$ FROM SAILORS INSERT INTO EMPLOYEE IF done THEN EXISTS examDB; VARCHAR(50), GRANT SELECT ON bill TO ADD COLUMN PINCODE INT EOREIGN KEY (bid Cust city VARCHAR2(20) (EmpNo. Name USE example: VARCHAR(50). NOT NULL; REFERENCES BOATS(bid) WHERE age = (SELECT -> salary INT END: ank\_name, Address MIN(age) FROM SAILORS); CREATE TABLE IF NOT last\_name 3) CREATE TABLE iv)ALTER TABLE Bank\_location END IF: SHOW GRANTS FOR -> ): INSERT INTO CUSTOMER EXISTS Hospital VARCHAR(50), DEPARTMENT Branch name, Email) INSERT INTO SAILORS (sid (Cust\_name, Cust\_street -> \$\$ Name VARCHAR(16) -> salary INT ID INT PRIMARY KEY CHANGE COLUMN DNAME 2) i)SELECT Cust city) VALUES ID'; sname, rating, age) REVOKE SELECT ON bill INSERT INTO employee INSERT INTO EMPLOYEES COUNT(DISTINCT sname) AS ->); VALUES (1, 'John Doe', 'ABC Bank' END LOOP Name VARCHAR(50). VALUES (ID, Name, Email) VALUES VALUES VARCHAR(255); CountOfSailorName VARCHAR(6) '123 Main St'. 'Citv1'. -> 55 (1. 'John Doe'. CLOSE cur ('John Doe', '123 Main St', (1, 'John', 8, 25), -> (1, 'John', 'Doe' v)ALTER TABLE 'john.doe@example.com' Hospital VARCHAR(6) 'City1'), 60000). INSERT INTO employees 'john.doe@example.com'), END // DEPARTMENT (2, 'Alice', 7, 22), ii)SELECT rating, AVG(age) VALUES SELECT \* FROM ('Alice Smith', '456 Oa -> (2, 'Jane', 'Smith', MODIFY COLUMN (2 'Alice Smith' 'YYZ DELIMITER CREATE TABLE AS AverageAge (3, 'Bob', 6, 30). St', 'City2'), Bank', '456 Oak St', 'City2' INSERT INTO Hospital -> (1. 'John', 'Doe EMPLOYEES AUDIT ( LOCATION CHAR(10): FROM SALLORS CALL GetEmailtds(): 60000), (4, 'Eva', 9, 21); UPDATE EMPLOYEES SET 'Branch2'. -> (3, 'Bob', 'Johnson' AuditID IN Name = 'John Updated 'alice.smith@example.com') INSERT INTO BOATS (bi Hospital) VALUES GROUP BY rating; St', 'City3'); -> (2, 'Jane', 'Smith' AUTO INCREMENT DEPARTMENT; WHERE ID = 1; ('John Doe', 'Dr. Smith' PRIMARY KEY, bname, color) ii)ALTER TABLE CUSTOME -> (4, 'Alice', 'Williams', SELECT \* FROM (3, 'Bob Johnson', 'PQR 'City Hospital'). VALUES 35000): -> (3, 'Bob', 'Johnson' EmployeeID INT. INSERT INTO SAILORS (sid. ADD COLUMN salary INT Bank', '789 Pine St', 'City3' EMPLOYEES: (iii) Alter the table column -> \$\$ (101 'Boat1' 'red') sname, rating, age) 'Branch3' OldName VARCHAR(50), 'Community Hospital'): domain

2)CREATE TABLE Course(CourseID) DepartmentID IN CREATE TABLE Student ( PRIMARY KEY, DenartmentNam StudentID INT PRIMARY VARCHAR(50) NOT NULL Name VARCHAR(50) NOT NULL CREATE TABLE Faculty ( FacultyID INT PRIMARY KEY, CREATE TABLE Score ( Name VARCHAR(50) NOT StudentID INT, NULL. CourseID INT, DepartmentID INT Score INT, FOREIGN KEY PRIMARY KEY (StudentID, (DepartmentID) CourseID) REFERENCES FOREIGN KEY (StudentID) REFERENCES CREATE TABLE Course ( FOREIGN KEY (CourseID) REFERENCES CourseID INT PRIMAR KEY, Course(CourseID) VARCHAR(10) NOT NULL INSERT INTO Department VARCHAR(50) NOT NULL DepartmentName) VALUES CREATE TABLE (1, 'ECE'), FacultyCourse ( (2, 'CS'), FacultyID INT (3, 'Math'); CourseID INT. INSERT INTO Faculty PRIMARY KEY (FacultyID (FacultyID, Name, CourseID). Department(D) FOREIGN KEY (FacultyID) VALUES

FOREIGN KEY (CourseID)

(1, 'John Smith', 1),

(2 'Alice Johnson' 2)

REFERENCES

SET 9

REFERENCES

Faculty(FacultyID)

INSERT INTO Course CourseName) VALUES (101 'CSC101' Programming'), (102, 'ECE201', 'Digital Electronics'). (103, 'MATH301 'Calculus'): INSERT INTO FacultyCourse (FacultyID, CourseID) VALUES (2, 102), (3, 101): INSERT INTO Studen (StudentID, Name) VALUES (1. 'Mark Green').SELECT FROM Faculty INNER JOIN Department ON Faculty DepartmentID = Department.DepartmentID INNER IOIN FacultyCourse ON Faculty.FacultyID = FacultyCourse FacultyID FacultyCourse.CourseID = WHERE Department.DepartmentNa me = 'ECE';

(3, 'Bob Doe', 1);

(3, 'Chris Black'); SELECT DISTINCT FROM Faculty INNER JOIN Department ON Faculty.DepartmentID = Department.DepartmentID; SELECT Student Name Course.CourseNumber FROM Student CROSS JOIN Course LEFT JOIN Score ON Student.StudentID = Score.StudentID AND Course.CourseID = Score.CourseID WHERE Score.StudentID IS NUIII -CREATE TABLE Department DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(50) NOT NULL CREATE TABLE Faculty ( FacultyID INT PRIMARY Name VARCHAR(50) NOT NULL, DepartmentID INT

FOREIGN KEY

REFERENCES

CREATE TABLE Course ( REFERENCES CourselD INT PRIMARY Course(CourseID) CourseNumber VARCHAR(10) NOT NULL, (DepartmentID. DenartmentName VALUES (1, 'ECE'), CREATE TABLE (2, 'CS'), FacultyCourse FacultyID INT, (3, 'Math'); CourseID INT, (FacultyID, Name PRIMARY KEY (FacultyID, DepartmentID) CourseID), VALUES FOREIGN KEY (FacultyID) REFERENCES Faculty(FacultyID) FOREIGN KEY (CourseID) (3. 'Bob Doe', 1): REFERENCES INSERT INTO Course CREATE TABLE Student ( VALUES StudentID INT PRIMARY (101, 'CSC101' KEY. Programming'), NULL Electronics'), CREATE TABLE Score ( (103 'MATH301 'Calculus'); StudentID INT, CourseID INT, Score INT, VALUES PRIMARY KEY (StudentID, (1, 101). Department(DepartmentID) (2, 102), FOREIGN KEY (StudentID) (3, 101) REFERENCES Student(StudentID)

FOREIGN KEY (CourseID) INSERT INTO Departmen INSERT INTO Faculty (1, 'John Smith', 1), (2, 'Alice Johnson', 2), (CourseID CourseNumber (102. 'ECE201', 'Digital INSERT INTO FacultyCours (FacultyID, CourseID)

INSERT INTO Studen 3) CREATE TABLE (StudentID, Name) EMPLOYEE ( EmpNo INT PRIMARY KEY (1. 'Mark Green'). Name VARCHAR(50) (2, 'Emily White'), VARCHAR(50) (3, 'Chris Black'); Address VARCHAR(100), SELECT DISTINCT DepartmentName VARCHAR(50). INNER JOIN Department ON VARCHAR(50), Faculty.DepartmentID = Email VARCHAR(50) Department.DepartmentID; SELECT Student.Nam Course.CourseNumber INSERT INTO EMPLOYEE Bank name, Address, CROSS JOIN Course Bank location, LEFT JOIN Score ON Branch\_name, Email) Student StudentID = VALUES ore.StudentID AND (1, 'John Doe', 'ABC Bank', Course.CourseID = '123 Main St', 'City1', WHERE Score.StudentID IS 'john.doe@example.com'), SELECT Course.CourseName Bank', '456 Oak St', 'Citv2' 'alice.smith@example.com') INNER IOIN Department ON Faculty.DepartmentID = Department,DepartmentID Bank' '789 Pine St' 'City3' ON Faculty FacultyID = 'bob.johnson@example.co FacultyCourse.FacultyID INNER JOIN Course ON DELIMITER // FacultyCourse.CourseID = CREATE PROCEDUR Course.CourseID GetEmailIds() Department.DepartmentNa

VALUES

FROM Faculty

FROM Student

Score CourseID

me = 'ECE';

DECLARE done INT DEFAULT FALSE; DECLARE email VARCHAR(50); DECLARE cur CURSOR SELECT Email FROM EMPLOYEE; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done - TRUE OPEN curread\_loop: LOOF FETCH cur INTO email; IF done THEN LEAVE read loop; END IF: SELECT email AS 'Email ın'-END LOOP CLOSE cur; FND // DELIMITER -CALL GetEmailIds(); 4) CREATE TABLE Student ( StudentID INT PRIMARY Name VARCHAR(50) NOT NULL, Email VARCHAR(100) NOT NULL.

Credits INT. CREATE TABLE Enrollment ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, EnrollmentDate DATE Grade VARCHAR(2) REFERENCES Student(StudentID) REFERENCES Course(CourseID) CREATE TABLE Instructor ( KEY NILILI NULL, Departmen VARCHAR(50) habbaan ŀ

CREATE TABLE Department CREATE TABLE Course ( CourseID INT PRIMAR PRIMARY KEY, KEY. CourseName VARCHAR(50) NOT NULL. VARCHAR(100) NOT NULL, CourseCode VARCHAR(50), VARCHAR(20) NOT NULL CREATE TABLE CourseInstructor CourseID INT. InstructorID INT PRIMARY KEY (CourseID, InstructorID). FOREIGN KEY (CourseID) REFERENCES FOREIGN KEY FOREIGN KEY (StudentID) (InstructorID) REFERENCES Instructor(InstructorID) FOREIGN KEY (CourseID) INSERT INTO Student (StudentID, Name, Email, DateOfBirth) VALUES (1, 'John Doe 'iohn.doe@example.com 1990-01-01'): InstructorID INT PRIMARY INSERT INTO Course CourseID. CourseNam Name VARCHAR(50) NOT CourseCode, Credits) VALUES (101, 'Introduction Email VARCHAR(100) NOT to SQL', 'SQL101', 3); INSERT INTO Enrollmen (EnrollmentID, StudentID CourseID. EnrollmentDate -- Add other attributes as Grade)

DECLARE cur CURSOR SELECT Email FROM DECLARE CONTINUE HANDLER FOR NOT FOUND FETCH cur INTO email LEAVE read loop: SELECT email AS 'Emai VALUES (1, 1, 101, '2023-01-15', 'A'); DESCRIBE Student

DESCRIBE Course:

DESCRIBE Enrollment;

DESCRIBE Department

DESCRIBE CourseInstructor

SET 10	SET iteration = iteration	(2, 'Alice Smith', 'XYZ	CALL GetEmailIds();	CREATE TABLE Department	FOREIGN KEY (PatientID)
2) i)DELIMITER //	+ 1;	Bank', '456 Oak St', 'City2', 'Branch2',		(	REFERENCES Patient(PatientID),
CREATE PROCEDURE	UNTIL income >= 4000 END REPEAT;	'alice.smith@example.com')		DepartmentID INT PRIMARY KEY,	FOREIGN KEY (DoctorID)
BuildStringUntilGreaterTha nFive()	SELECT iteration AS	(3. 'Bob Johnson', 'POR		DepartmentName	REFERENCES Doctor(DoctorID)
BEGIN	IterationCount, income AS FinalIncome:	Bank', '789 Pine St', 'City3',	4) CREATE TABLE Patient (	VARCHAR(50) NOT NULL,	);
DECLARE str	END //	'Branch3', 'bob.johnson@example.co	PatientID INT PRIMARY KEY.	Description VARCHAR(255)	DESCRIBE Patient:
VARCHAR(255) DEFAULT ";	DELIMITER:	m');	Name VARCHAR(50) NOT	);	DESCRIBE Pacient,
DECLARE counter INT DEFAULT 1;	to get op type->CALL	DELIMITER //	NULL,	CREATE TABLE Appointment	DESCRIBE Nurse:
REPEAT	RepeatUntilIncomeGreater	CREATE PROCEDURE	DateOfBirth DATE,	(	DESCRIBE Department;
SET str = CONCAT(str.	OrEqual4000();	GetEmailIds() REGIN	Gender VARCHAR(10),	AppointmentID INT	DESCRIBE Appointment;
'Iteration ', counter, ' ');		DECLARE done INT	ContactNumber	PRIMARY KEY,	DESCRIBE MedicalRecord;
SET counter = counter +	3) CREATE TABLE EMPLOYEE (	DEFAULT FALSE;	VARCHAR(20)	PatientID INT,	,
1;	EmpNo INT PRIMARY KEY,	DECLARE email	):	DoctorID INT,	
UNTIL counter > 5 END REPEAT;	Name VARCHAR(50),	VARCHAR(50);	CREATE TABLE Doctor (	AppointmentDate DATETIME,	
SELECT str AS FinalString;	Bank_name	DECLARE cur CURSOR FOR	DoctorID INT PRIMARY KEY,	Notes TEXT,	
END //	VARCHAR(50),	SELECT Email FROM	Name VARCHAR(50) NOT	FOREIGN KEY (PatientID)	
DELIMITER;	Address VARCHAR(100),	EMPLOYEE;	NULL,	REFERENCES Patient(PatientID),	
to get op type ->CALL	Bank_location VARCHAR(50),	DECLARE CONTINUE	Specialization VARCHAR(50).	FOREIGN KEY (DoctorID)	
BuildStringUntilGreaterTha nFive();	Branch name	HANDLER FOR NOT FOUND SET done = TRUE;	ContactNumber	REFERENCES	
ii)DELIMITER //	VARCHAR(50),	OPEN cur;	VARCHAR(20)	Doctor(DoctorID)	
CREATE PROCEDURE	Email VARCHAR(50)	read_loop: LOOP	);	);	
RepeatUntilIncomeGreater	);	FETCH cur INTO email;	CREATE TABLE Nurse (	CREATE TABLE MedicalRecord (	
OrEqual4000() REGIN	INSERT INTO EMPLOYEE (EmpNo. Name.	IF done THEN	NurseID INT PRIMARY	RecordID INT PRIMARY	
DECLARE income INT	Bank_name, Address,	LEAVE read_loop;	KEY,	KEY,	
DEFAULT 0;	Bank_location, Branch name, Email)	END IF;	Name VARCHAR(50) NOT NULL,	PatientID INT,	
DECLARE iteration INT	VALUES	SELECT email AS 'Email	Department	DoctorID INT,	
DEFAULT 1;	(1. 'John Doe', 'ABC Bank'.	ID';	VARCHAR(50),	Diagnosis TEXT,	
REPEAT	'123 Main St', 'City1',	END LOOP;	ContactNumber VARCHAR(20)	Treatment TEXT,	
SET income = income + 500: Increment income	'Branch1', 'john.doe@example.com'),	CLOSE cur;	);	Prescription TEXT,	
(for a )		END //	Ir		

DELIMITER;