**ASSIGNMENT [ TASK 1, TASK 2, TASK 3]**

CREATE DATABASE SISDB ;

USE SISDB ;

CREATE TABLE Students

( Student\_id INT AUTO\_INCREMENT PRIMARY KEY ,

First\_name VARCHAR(50) ,

Last\_name VARCHAR(50) ,

Date\_of\_birth DATE NOT NULL ,

Email VARCHAR(50) UNIQUE,

Phone\_no VARCHAR(10)

);

CREATE TABLE Teachers

( Teacher\_id INT AUTO\_INCREMENT PRIMARY KEY ,

First\_name VARCHAR (50),

Last\_name VARCHAR (50),

Email VARCHAR (50) UNIQUE

);

CREATE TABLE Courses

( Course\_id INT AUTO\_INCREMENT PRIMARY KEY,

Course\_name VARCHAR(20) ,

Credits INT NOT NULL ,

Teacher\_id INT ,

FOREIGN KEY (Teacher\_id) REFERENCES Teachers (Teacher\_id)

);

CREATE TABLE Enrollments

( enrollment\_id INT AUTO\_INCREMENT PRIMARY KEY ,

Student\_id INT ,

Course\_id INT ,

Enrollment\_date DATE ,

FOREIGN KEY (Student\_id) REFERENCES Students (Student\_id ) ,

FOREIGN KEY (Course\_id) REFERENCES Courses ( Course\_id)

);

CREATE TABLE Payments

( Payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

Student\_id INT ,

Amount DECIMAL (10 , 2) DEFAULT 10000,

Payment\_date DATE ,

FOREIGN KEY (Student\_id) REFERENCES Students (Student\_id )

);

-- Insert records into Students table

INSERT INTO Students (Student\_id, First\_name, Last\_name, Date\_of\_birth, Email, Phone\_no)

VALUES

(1, 'Vidushi', 'Upadhyay', '2002-08-01', 'vidushiUPAD123@gmail.com', '9174162097'),

(2, 'Sneha', 'Upadhyay', '1997-07-05', 'snehaupadyay12@gmail.com', '9987654321'),

(3, 'Vedic', 'Upadhyay', '2003-12-20', 'vedic123@gmail.com', '9564537865'),

(4, 'Anaya', 'Joshi', '2020-01-05', 'anayajoshi765@gmail.com', '3456789012'),

(5, 'Vaibhav', 'Joshi', '2000-05-15', 'vjoshi54@gmail.com', '9357890123'),

(6, 'Anuj', 'Tripathi', '1999-06-25', 'tripathianuj123@gmail.com', '9678901234'),

(7, 'Aditi', 'Tripathi', '2001-07-10', 'adititripathi65@gmail.com', '9789012345'),

(8, 'Vedant', 'Pandey', '2002-08-20', 'vedant345@gmail.com', '7890123456'),

(9, 'Vaishali', 'Pandey', '2000-09-05', 'vaishalip987@gmail.com', '8901234567'),

(10, 'Vivek', 'Mishra', '1999-10-12', 'vivekmis54@gmail.com', '9012345678');

INSERT INTO Teachers VALUES

(21, 'Rohan', 'Sharma', 'rohan.sharma@gmail.com'),

(22, 'Anjali', 'Verma', 'anjali.verma@gmail.com'),

(23, 'Vikas', 'Gupta', 'vikas.gupta@gmail.com'),

(24, 'Sunita', 'Singh', 'sunita.singh@gmail.com'),

(25, 'Rajesh', 'Yadav', 'rajesh.yadav@gmail.com'),

(26, 'Meena', 'Kumari', 'meena.kumari@gmail.com'),

(27, 'Amit', 'Chaudhary', 'amit.chaudhary@gmail.com'),

(28, 'Neha', 'Mishra', 'neha.mishra@gmail.com'),

(29, 'Pooja', 'Pandey', 'pooja.pandey@gmail.com'),

(30, 'Deepak', 'Jain', 'deepak.jain@gmail.com');

INSERT INTO COURSES VALUES

(101, 'Mathematics', 4, 21),

(102, 'Science', 3, 22),

(103, 'History', 3, 23),

(104, 'English', 4, 24),

(105, 'Geography', 3, 25),

(106, 'Biology', 4, 26),

(107, 'Physics', 3, 27),

(108, 'Chemistry', 4, 28),

(109, 'Art', 2, 29),

(110, 'Computer Science', 3, 30);

INSERT INTO Enrollments

VALUES

(51, 2, 106, '2024-01-02'),

(52, 3, 107, '2024-01-05'),

(53, 7, 105, '2024-01-08'),

(54, 1, 104, '2024-01-12'),

(55, 5, 103, '2024-01-14'),

(56, 10, 110, '2024-01-18'),

(57, 4, 102, '2024-01-20'),

(58, 6, 108, '2024-01-25'),

(59, 8, 108, '2024-01-28'),

(60, 9, 101, '2024-01-30');

INSERT INTO Payments

VALUES

(301, 1, 20000.00, '2024-01-15'),

(302, 2, 33000.00, '2024-01-22'),

(303, 3, 23000.00, '2024-01-30'),

(304, 4, 55000.00, '2024-02-02'),

(305, 5, 66000.00, '2024-02-10'),

(306, 6, 12000.00, '2024-02-15'),

(307, 7, 19000.00, '2024-02-20'),

(308, 8, 22000.00, '2024-03-01'),

(309, 9, 70000.00, '2024-03-05'),

(310, 10, 13000.00, '2024-03-12');

-- TASK 2.1

INSERT INTO Students (Student\_id, First\_name, Last\_name, Date\_of\_birth, Email, Phone\_no)

VALUES (11 ,'John', 'Doe', '2000-01-01', 'john.doe@example.com', '1234567890');

-- TASK 2.2

INSERT INTO Enrollments (enrollment\_id, Student\_id, Course\_id, Enrollment\_date)

VALUES (61, 11, 101, '2024-03-15');

-- TASK 2.3

UPDATE Teachers

SET Email = 'manju432@gmail.com'

WHERE Teacher\_id = '21' ;

-- TASK 2.4

DELETE FROM Enrollments

WHERE Student\_id = 11 AND Course\_id = 101;

-- TASK 2.5

UPDATE Courses

SET Teacher\_id= 28

WHERE Course\_id = 101 ;

-- TASK 2.6

DELETE FROM Enrollments

WHERE Student\_id = '3';

DELETE FROM Payments

WHERE Student\_id = 3;

DELETE FROM Students

WHERE Student\_id = '3';

-- TASK 2.7

UPDATE Payments

SET Amount = 35000.00

WHERE Payment\_id = 302;

-- TASK 3.1

SELECT Students.First\_name, Students.Last\_name, SUM(Payments.Amount) AS Total\_Payments

FROM Payments

INNER JOIN Students ON Payments.Student\_id = Students.Student\_id

WHERE Students.Student\_id = 1

GROUP BY Students.First\_name, Students.Last\_name;

-- TASK 3.2

SELECT C.Course\_id, C.Course\_name, COUNT(E.Student\_id) AS Student\_Count

FROM Courses C

LEFT JOIN Enrollments E ON C.Course\_id = E.Course\_id

GROUP BY C.Course\_id, C.Course\_name

ORDER BY C.Course\_id;

-- TASK 3.3

SELECT Students.First\_name, Students.Last\_name

FROM Students

LEFT JOIN Enrollments ON Students.Student\_id = Enrollments.Student\_id

WHERE Enrollments.Student\_id IS NULL;

-- TASK 3.4

SELECT Students.First\_name, Students.Last\_name, Courses.Course\_name

FROM Students

INNER JOIN Enrollments ON Students.Student\_id = Enrollments.Student\_id

INNER JOIN Courses ON Enrollments.Course\_id = Courses.Course\_id;

-- TASK 3.5

SELECT Teachers.First\_name, Teachers.Last\_name, Courses.Course\_name

FROM Teachers

INNER JOIN Courses ON Teachers.Teacher\_id = Courses.Teacher\_id;

-- TASK 3.6

SELECT Students.First\_name, Students.Last\_name, Enrollments.Enrollment\_date

FROM Students

INNER JOIN Enrollments ON Students.Student\_id = Enrollments.Student\_id

INNER JOIN Courses ON Enrollments.Course\_id = Courses.Course\_id;

-- TASK 3.7

SELECT Students.First\_name, Students.Last\_name

FROM Students

LEFT JOIN Payments ON Students.Student\_id = Payments.Student\_id

WHERE Payments.Student\_id IS NULL;

-- TASK 3.8

SELECT Courses.Course\_name

FROM Courses

LEFT JOIN Enrollments ON Courses.Course\_id = Enrollments.Course\_id

WHERE Enrollments.Course\_id IS NULL;

-- TASK 3.9

SELECT E1.Student\_id, COUNT(E1.Course\_id) AS CourseCount

FROM Enrollments E1

JOIN Enrollments E2 ON E1.Student\_id = E2.Student\_id AND E1.enrollment\_id <> E2.enrollment\_id

GROUP BY E1.Student\_id

HAVING COUNT(E1.Course\_id) > 1;

-- TASK 3.10

SELECT Teachers.First\_name, Teachers.Last\_name

FROM Teachers

LEFT JOIN Courses ON Teachers.Teacher\_id = Courses.Teacher\_id

WHERE Courses.Teacher\_id IS NULL;