**Step 1**

CREATE DATABASE uni\_tasks;

CREATE TABLE Students(

student\_id INT(60) AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(50),

date\_of\_birth DATE,

gender ENUM ('Male', 'Female'),

major VARCHAR(50),

enrollment\_year INT(60));

CREATE TABLE Courses(

course\_id INT(60) AUTO\_INCREMENT PRIMARY KEY,

course\_name VARCHAR(100) NOT NULL,

course\_code VARCHAR(50) NOT NULL,

credits VARCHAR(50),

department VARCHAR(50));

CREATE TABLE Instructors(

instructor\_id INT(60) AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(50),

hire\_date DATE,

department VARCHAR(50));

CREATE TABLE Enrollments(

enrollment\_id INT(60) AUTO\_INCREMENT PRIMARY KEY,

student\_id INT(50),

course\_id INT(50),

grade INT(100)

);

CREATE TABLE Course\_Assignments(

assignment\_id INT(60) AUTO\_INCREMENT PRIMARY KEY,

instructor\_id INT(60) NOT NULL,

course\_id INT(60) NOT NULL,

semester VARCHAR(50),

course\_year INT(60)

);

**Step 2**

INSERT INTO `students` (`student\_id`, `first\_name`, `last\_name`, `email`, `date\_of\_birth`, `gender`, `major`, `enrollment\_year`) VALUES (NULL, 'Zaher', 'G.A', 'G.Zaher@gmail.com', '2001-11-07', 'Male', 'Civil Engineer', '2019'), (NULL, 'Rama', 'S.A', 'S.Rama@gmail.com', '2000-02-17', 'Female', 'Accounting', '2001'), (NULL, 'Kareem', 'C.L', 'C.Kareem@gmail.com', '1999-10-07', 'Male', 'Civil Engineer', '2019'),(NULL, 'Bushra', 'L.O', 'L.Bushra@gmail.com', '2001-09-09', 'Female', 'Computer Science', '2019'),(NULL, 'Malak', 'G.E', 'G.Malak@gmail.com', '2002-12-24', 'Female', 'Computer Science', '2019'),(NULL, 'Yazeed', 'N.O', 'N.Yazeed@gmail.com', '2001-10-21', 'Male', 'Software', '2019'),(NULL, 'Noor', 'O.P', 'O.Noor@gmail.com', '1998-06-19', 'Female', 'Civil engineer', '2019'),(NULL, 'Elina', 'R.O', 'R.Elina@gmail.com', '2000-12-07', 'Female', 'Software', '2019'),(NULL, 'Khaleel', 'B.E', 'B.Khaleel@gmail.com', '2000-07-27', 'Male', 'Software', '2019');

INSERT INTO `instructors`(`instructor\_id`, `first\_name`, `last\_name`, `email`, `hire\_date`, `department`) VALUES (Null,'Damer','A.W','A.Damer@gmail.com','2002-02-02','Engineer'),(Null,'Suhair','F.C','F.Suhair@gmail.com','2015-01-11','Engineer'),(Null,'Saber','A.B','A.Saber@gmail.com','2005-07-01','Accounting'),(Null,'Lana','R.O','A.Lana@gmail.com','2011-03-03','IT'),(Null,'Mohammed','S.N','S.Mohammed@gmail.com','2009-08-11','IT');

INSERT INTO `courses`(`course\_id`, `course\_name`, `course\_code`, `credits`, `department`) VALUES (Null,'Structure','1222','3h','Engineering'),(Null,'English','1111','3h','Languege'),(Null,'Accounting01','1503','3h','Accounting'),(Null,'Database','1331','4h','IT'),(Null,'Software','1321','3h','IT');

INSERT INTO `Course\_assignments` (`instructor\_id`, `course\_id`, `semester`, `course\_year`)

VALUES (1, 1, 'First', 2023), (2, 2, 'First', 2023),(3, 3, 'Second', 2023), (4, 4, 'Second', 2023), (5, 5, 'Second', 2023);

INSERT INTO `Enrollments` (`student\_id`, `course\_id`, `grade`)

VALUES

(1, 1, 'D'), (1, 5, 'C'), (2, 3, 'A'),(2, 2, 'A'), (3, 1, 'B'), (3, 4, 'B'), (4, 4, 'C'), (4, 5, 'D'), (5, 4, 'C'), (5, 5, 'A'), (6, 5, 'B'), (6, 4, 'D'),(7, 1, 'B'), (7, 2, 'A'), (8, 5, 'C'), (8, 4, 'A'),(9, 5, 'C'),(9, 4, 'B');

**Step 3:**

**Queries Basic Queries:**

SELECT \* FROM `students`

SELECT COUNT(\*) AS total\_courses FROM `courses`;

SELECT s.first\_name, s.last\_name FROM `students` s JOIN `Enrollments` e ON s.student\_id = e.student\_id WHERE e.course\_id = 1;

SELECT `email` FROM `instructors` WHERE `department` = 'IT';

**Intermediate Queries:**

SELECT c.course\_name, COUNT(e.student\_id) AS student\_count

FROM `courses` c

JOIN `Enrollments` e ON c.course\_id = e.course\_id

GROUP BY c.course\_id, c.course\_name;

SELECT s.first\_name, s.last\_name, e.course\_id, e.grade

FROM `students` s

JOIN `Enrollments` e ON s.student\_id = e.student\_id

WHERE e.grade = 'A';

SELECT c.course\_name, i.first\_name, i.last\_name, ca.semester

FROM `courses` c

JOIN `course\_assignments` ca ON c.course\_id = ca.course\_id

JOIN `instructors` i ON ca.instructor\_id = i.instructor\_id

WHERE ca.semester = 'First';

SELECT AVG(e.grade) AS average\_grade FROM `Enrollments` e WHERE e.course\_id = 1;

**Advanced Queries:**

SELECT s.first\_name, s.last\_name, COUNT(e.course\_id)

FROM `students` s

JOIN `Enrollments` e ON s.student\_id = e.student\_id

JOIN `course\_assignments` ca ON e.course\_id = ca.course\_id

WHERE ca.semester = 'Second' AND ca.course\_year = 2023

GROUP BY s.student\_id, s.first\_name, s.last\_name

HAVING COUNT(e.course\_id) > 3;

SELECT s.first\_name, s.last\_name, e.course\_id, e.grade

FROM `students` s

JOIN `Enrollments` e ON s.student\_id = e.student\_id

WHERE e.grade != 'A';

SELECT s.first\_name, s.last\_name, AVG(

CASE

WHEN e.grade = 'A' THEN 4

WHEN e.grade = 'B' THEN 3

WHEN e.grade = 'C' THEN 2

WHEN e.grade = 'D' THEN 1

WHEN e.grade = 'F' THEN 0

ELSE NULL

END

) AS max\_grade

FROM `students` s

JOIN `Enrollments` e ON s.student\_id = e.student\_id

GROUP BY s.student\_id, s.first\_name, s.last\_name

ORDER BY max\_grade DESC

LIMIT 1;

SELECT c.department, COUNT(c.course\_id) AS course\_count

FROM `courses` c

JOIN `course\_assignments` ca ON c.course\_id = ca.course\_id

WHERE ca.course\_year = 2023

GROUP BY c.department

ORDER BY course\_count DESC

LIMIT 1;

SELECT c.course\_id, c.course\_name, COUNT(e.student\_id) AS count\_student

FROM `courses` c

LEFT JOIN `Enrollments` e ON c.course\_id = e.course\_id

GROUP BY c.course\_id, c.course\_name

HAVING count\_student = 0;

**Step 4: Functions and Aggregates**

**function**

DELIMITER $$

CREATE FUNCTION calculate\_age(year\_of\_birth INT)

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE age INT;

SET age = YEAR(CURDATE()) - year\_of\_birth;

RETURN age;

END $$

DELIMITER ;

**sql**

SELECT first\_name, last\_name, calculate\_age(YEAR(date\_of\_birth)) AS age

FROM `students`;

**function**

DELIMITER $$

CREATE PROCEDURE enroll\_student(

IN p\_student\_id INT,

IN p\_course\_id INT,

IN p\_grade VARCHAR(1) )

BEGIN

INSERT INTO `Enrollments` (student\_id, course\_id, grade)

VALUES (p\_student\_id, p\_course\_id, p\_grade);

END $$

DELIMITER ;

**sql**

CALL enroll\_student(1, 2, 'A');

SELECT

s.major AS department,

AVG(CASE

WHEN e.grade = 'A' THEN 4.0

WHEN e.grade = 'B' THEN 3.0

WHEN e.grade = 'C' THEN 2.0

WHEN e.grade = 'D' THEN 1.0

WHEN e.grade = 'F' THEN 0.0

ELSE NULL

END) AS average\_grade

FROM

Enrollments e

JOIN

students s ON e.student\_id = s.student\_id

GROUP BY

s.major;

**Step 5: Constraints and Transactions**

SELECT email, COUNT(\*)

FROM students

GROUP BY email

HAVING COUNT(\*) > 1;

ALTER TABLE students

ADD CONSTRAINT unique\_email UNIQUE (email);