Create database and tables:-

```sql
create database student_db;

Use student_db;


-- 1. Departments
CREATE TABLE Departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(100) NOT NULL
);

-- 2. Students
CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    dob DATE,
    gender VARCHAR(10),
    email VARCHAR(100) UNIQUE,
    phone_number VARCHAR(15),
    address TEXT,
    admission_date DATE,
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES Departments(department_id)
);


-- 3. Faculty
CREATE TABLE Faculty (
    faculty_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE,
    phone_number VARCHAR(15),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES Departments(department_id)
);

-- 4. Courses
CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(100) NOT NULL,
    faculty_id INT,
    FOREIGN KEY (faculty_id) REFERENCES Faculty(faculty_id)
);
```

```sql
-- 5. Enrollments
CREATE TABLE Enrollments (
    enrollment_id INT 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)
);

-- 6. Attendance
CREATE TABLE Attendance (
    attendance_id INT PRIMARY KEY,
    student_id INT,
    course_id INT,
    attendance_date DATE,
    status VARCHAR(10),
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);


-- 7. Grades
CREATE TABLE Grades (
    grade_id INT PRIMARY KEY,
    student_id INT,
    course_id INT,
    marks_obtained INT,
    grade VARCHAR(5),
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```

Insert Records:-

```
INSERT INTO Departments(department_id,department_name) VALUES
(1, 'Computer Science'),
(2, 'Information Technology'),
(3, 'Electronics'),
(4, 'Mechanical');

INSERT INTO Students
(student_id, name, dob, gender, email, phone_number, address, admission_date, department_id)
VALUES
(101, 'Amit Sharma', '2002-05-10', 'Male', 'amit@gmail.com', '9876543210', 'Ahmedabad', '2021-07-10', 1),

(102, 'Neha Patel', '2003-02-15', 'Female', NULL, '9876543211', 'Surat', '2022-06-15', 1),

(103, 'Rahul Verma', '2001-09-20', 'Male', 'rahul@gmail.com', '9876543212', 'Vadodara', '2020-08-01', 2),

(104, 'Priya Singh', '2004-01-05', 'Female', 'priya@gmail.com', '9876543213', 'Rajkot', '2023-01-20', 3),

(105, 'Karan Mehta', '2002-11-25', 'Male', NULL, '9876543214', 'Ahmedabad', '2021-09-12', 2),

(106, 'Sneha Joshi', '2000-03-18', 'Female', 'sneha@gmail.com', '9876543215', 'Pune', '2019-07-05', 1),

(107, 'Vikas Yadav', '2003-06-30', 'Male', 'vikas@gmail.com', '9876543216', 'Delhi', '2022-02-11', 4),

(108, 'Anjali Desai', '2004-08-12', 'Female', 'anjali@gmail.com', '9876543217', 'Mumbai', '2023-07-18', 3);
```

```
INSERT INTO Faculty
(faculty_id, name, email, phone_number, department_id)
VALUES
(201, 'Dr Mehta', 'mehta@college.com', '9000000001', 1),
(202, 'Prof Shah', 'shah@college.com', '9000000002', 2),
(203, 'Dr Iyer', 'iyer@college.com', '9000000003', 3),
(204, 'Prof Khan', 'khan@college.com', '9000000004', 4),
(205, 'Dr Patel', 'patel@college.com', '9000000005', 1);
```

```
INSERT INTO Courses VALUES
(301, 'Database Systems', 201),
(302, 'Operating Systems', 205),
(303, 'Data Structures', 202),
(304, 'Digital Electronics', 203),
(305, 'Thermodynamics', NULL);
```

```
INSERT INTO Enrollments VALUES
(401, 101, 301, '2023-01-10'),
(402, 102, 301, '2023-01-11'),
(403, 103, 303, '2023-01-15'),
(404, 104, 304, '2023-02-01'),
(405, 105, 303, '2023-02-10'),
(406, 106, 302, '2023-02-12'),
(407, 101, 302, '2023-03-01'),
(408, 107, 305, '2023-03-10');

INSERT INTO Attendance(attendance_id,student_id,course_id,attendance_date,status) VALUES
(501, 101, 301, '2023-03-01', 'Present'),
(502, 102, 301, '2023-03-02', 'Present'),
(503, 103, 303, '2023-03-03', 'Absent'),
(504, 104, 304, '2023-03-04', 'Present'),
(505, 105, 303, '2023-03-05', 'Absent'),
(506, 106, 302, '2023-03-06', 'Present'),
(507, 107, 305, '2023-03-07', 'Absent');



INSERT INTO Grades(grade_id,student_id,course_id,marks_obtained,grade) VALUES
(601, 101, 301, 95, 'A+'),
(602, 102, 301, 78, 'B+'),
(603, 103, 303, 45, 'D'),
(604, 104, 304, 88, 'A'),
(605, 105, 303, 67, 'C'),
(606, 106, 302, 91, 'A+'),
(607, 107, 305, 35, 'F');
```

All Tables Records:-

```
select * from Departments;
```

department_id	department_name
1	Computer Science
2	Information Technology
3	Electronics
4	Mechanical
NULL	NULL

```
select * from Students;
```

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	9876543210	Ahmedabad	2021-07-10	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
103	Rahul Verma	2001-09-20	Male	rahul@gmail.com	9876543212	Vadodara	2020-08-01	2
104	Priya Singh	2004-01-05	Female	priya@gmail.com	9876543213	Rajkot	2023-01-20	3
105	Karan Mehta	2002-11-25	Male	NULL	9876543214	Ahmedabad	2021-09-12	2
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
107	Vikas Yadav	2003-06-30	Male	vikas@gmail.com	9876543216	Delhi	2022-02-11	4
108	Anjali Desai	2004-08-12	Female	anjali@gmail.com	9876543217	Mumbai	2023-07-18	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
select * from Faculty;
```

faculty_id	name	email	phone_number	department_id
201	Dr Mehta	mehta@college.com	9000000001	1
202	Prof Shah	shah@college.com	9000000002	2
203	Dr Iyer	iyer@college.com	9000000003	3
204	Prof Khan	khan@college.com	9000000004	4
205	Dr Patel	patel@college.com	9000000005	1
NULL	NULL	NULL	NULL	NULL

```
select * from Courses;
```

course_id	course_name	faculty_id
301	Database Systems	201
302	Operating Systems	205
303	Data Structures	202
304	Digital Electronics	203
305	Thermodynamics	NULL
NULL	NULL	NULL

```
select * from Enrollments;
```

enrollment_id	student_id	course_id	enrollment_date
401	101	301	2023-01-10
402	102	301	2023-01-11
403	103	303	2023-01-15
404	104	304	2023-02-01
405	105	303	2023-02-10
406	106	302	2023-02-12
407	101	302	2023-03-01
408	107	305	2023-03-10
NULL	NULL	NULL	NULL

```
select * from Attendance;
```

attendance_id	student_id	course_id	attendance_date	status
501	101	301	2023-03-01	Present
502	102	301	2023-03-02	Present
503	103	303	2023-03-03	Absent
504	104	304	2023-03-04	Present
505	105	303	2023-03-05	Absent
506	106	302	2023-03-06	Present
507	107	305	2023-03-07	Absent
NULL	NULL	NULL	NULL	NULL

```
select * from  Grades;
```

grade_id	student_id	course_id	marks_obtained	grade
601	101	301	95	A+
602	102	301	78	B+
603	103	303	45	D
604	104	304	88	A
605	105	303	67	C
606	106	302	91	A+
607	107	305	35	F
NULL	NULL	NULL	NULL	NULL

SQL Queries:-

```
-- 1. Implement CRUD Operations (Low Weightage)

--   How can you insert new students, faculty members, courses, and enrollments?
Insert into Students values
(109,"Kunal Sharma",'2003-12-13','Male','kunal@gmail.com','9789897867','Ahmedabad','2021-08-12',1);
```

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	9876543210	Ahmedabad	2021-07-10	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
103	Rahul Verma	2001-09-20	Male	rahul@gmail.com	9876543212	Vadodara	2020-08-01	2
104	Priya Singh	2004-01-05	Female	priya@gmail.com	9876543213	Rajkot	2023-01-20	3
105	Karan Mehta	2002-11-25	Male	NULL	9876543214	Ahmedabad	2021-09-12	2
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
107	Vikas Yadav	2003-06-30	Male	vikas@gmail.com	9876543216	Delhi	2022-02-11	4
108	Anjali Desai	2004-08-12	Female	anjali@gmail.com	9876543217	Mumbai	2023-07-18	3
109	Kunal Sharma	2003-12-13	Male	kunal@gmail.com	9789897867	Ahmedabad	2021-08-12	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
Insert into Faculty values
(206,'Dr Karan','karan2@college.com','9000000012',1);
```

faculty_id	name	email	phone_number	department_id
201	Dr Mehta	mehta@college.com	9000000001	1
202	Prof Shah	shah@college.com	9000000002	2
203	Dr Iyer	iyer@college.com	9000000003	3
204	Prof Khan	khan@college.com	9000000004	4
205	Dr Patel	patel@college.com	9000000005	1
206	Dr Karan	karan2@college.com	9000000012	1
NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Courses VALUES
(306, 'Web Development', 206);
```

course_id	course_name	faculty_id
301	Database Systems	201
302	Operating Systems	205
303	Data Structures	202
304	Digital Electronics	203
305	Thermodynamics	NULL
306	Web Development	206
NULL	NULL	NULL

```
Insert into Enrollments values
(409, 109, 306, '2023-03-12');
```

enrollment_id	student_id	course_id	enrollment_date
401	101	301	2023-01-10
402	102	301	2023-01-11
403	103	303	2023-01-15
404	104	304	2023-02-01
405	105	303	2023-02-10
406	106	302	2023-02-12
407	101	302	2023-03-01
408	107	305	2023-03-10
409	109	306	2023-03-12
NULL	NULL	NULL	NULL

```
--   How can student records be updated when contact details change?
    Update students set phone_number="8878943218" where student_id=101;
```

Before Update

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	9876543210	Ahmedabad	2021-07-10	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
103	Rahul Verma	2001-09-20	Male	rahul@gmail.com	9876543212	Vadodara	2020-08-01	2
104	Priya Singh	2004-01-05	Female	priya@gmail.com	9876543213	Rajkot	2023-01-20	3
105	Karan Mehta	2002-11-25	Male	NULL	9876543214	Ahmedabad	2021-09-12	2
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
107	Vikas Yadav	2003-06-30	Male	vikas@gmail.com	9876543216	Delhi	2022-02-11	4
108	Anjali Desai	2004-08-12	Female	anjali@gmail.com	9876543217	Mumbai	2023-07-18	3
109	Kunal Sharma	2003-12-13	Male	kunal@gmail.com	9789897867	Ahmedabad	2021-08-12	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

After Update

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	8878943218	Ahmedabad	2021-07-10	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
103	Rahul Verma	2001-09-20	Male	rahul@gmail.com	9876543212	Vadodara	2020-08-01	2
104	Priya Singh	2004-01-05	Female	priya@gmail.com	9876543213	Rajkot	2023-01-20	3
105	Karan Mehta	2002-11-25	Male	NULL	9876543214	Ahmedabad	2021-09-12	2
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
107	Vikas Yadav	2003-06-30	Male	vikas@gmail.com	9876543216	Delhi	2022-02-11	4
108	Anjali Desai	2004-08-12	Female	anjali@gmail.com	9876543217	Mumbai	2023-07-18	3
109	Kunal Sharma	2003-12-13	Male	kunal@gmail.com	9789897867	Ahmedabad	2021-08-12	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
--    How can students who have dropped out be deleted from the database?
delete from students where student_id = 108;
```

Before delete

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	8878943218	Ahmedabad	2021-07-10	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
103	Rahul Verma	2001-09-20	Male	rahul@gmail.com	9876543212	Vadodara	2020-08-01	2
104	Priya Singh	2004-01-05	Female	priya@gmail.com	9876543213	Rajkot	2023-01-20	3
105	Karan Mehta	2002-11-25	Male	NULL	9876543214	Ahmedabad	2021-09-12	2
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
107	Vikas Yadav	2003-06-30	Male	vikas@gmail.com	9876543216	Delhi	2022-02-11	4
108	Anjali Desai	2004-08-12	Female	anjali@gmail.com	9876543217	Mumbai	2023-07-18	3
109	Kunal Sharma	2003-12-13	Male	kunal@gmail.com	9789897867	Ahmedabad	2021-08-12	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

After delete

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	8878943218	Ahmedabad	2021-07-10	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
103	Rahul Verma	2001-09-20	Male	rahul@gmail.com	9876543212	Vadodara	2020-08-01	2
104	Priya Singh	2004-01-05	Female	priya@gmail.com	9876543213	Rajkot	2023-01-20	3
105	Karan Mehta	2002-11-25	Male	NULL	9876543214	Ahmedabad	2021-09-12	2
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
107	Vikas Yadav	2003-06-30	Male	vikas@gmail.com	9876543216	Delhi	2022-02-11	4
109	Kunal Sharma	2003-12-13	Male	kunal@gmail.com	9789897867	Ahmedabad	2021-08-12	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- 2. Use SQL Clauses (WHERE, HAVING, LIMIT) (Low Weightage)

--  How can you get students enrolled in the Computer Science department?
    select * from students where department_id = 1;
```

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	8878943218	Ahmedabad	2021-07-10	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
109	Kunal Sharma	2003-12-13	Male	kunal@gmail.com	9789897867	Ahmedabad	2021-08-12	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
--  How can you retrieve the top 5 highest-scoring students?
    SELECT s.student_id,
       s.name,
       g.marks_obtained
    FROM Grades g
    JOIN Students s
    ON g.student_id = s.student_id
    ORDER BY g.marks_obtained DESC
    LIMIT 5;
```

student_id	name	marks_obtained
101	Amit Sharma	95
106	Sneha Joshi	91
104	Priya Singh	88
102	Neha Patel	78
105	Karan Mehta	67

```sql
--  How can you find students whose attendance is below 75%?
    SELECT
    s.student_id,
    s.name,
    (SUM(a.status = 'Present') * 100 / COUNT(*)) AS attendance_percentage
    FROM Attendance a
    JOIN Students s
    ON a.student_id = s.student_id
    GROUP BY s.student_id, s.name
    HAVING attendance_percentage < 75;
```

	student_id	name	attendance_percentage
▶	103	Rahul Verma	0.0000
	105	Karan Mehta	0.0000
	107	Vikas Yadav	0.0000

```sql
-- 3. Apply SQL Operators (AND, OR, NOT) (Medium Weightage)

--  How can you retrieve students who have attendance below 50% AND are failing?
    SELECT
    s.student_id,
    s.name,
    ROUND((SUM(a.status = 'Present') * 100.0) / COUNT(a.attendance_id), 2)
        AS attendance_percentage,
    SUM(g.marks_obtained) AS total_marks
    FROM Students s
    JOIN Attendance a
    ON s.student_id = a.student_id
    JOIN Grades g
    ON s.student_id = g.student_id
    GROUP BY s.student_id, s.name
    HAVING attendance_percentage < 50
    AND total_marks < 40;
```

	student_id	name	attendance_percentage	total_marks
▶	107	Vikas Yadav	0.00	35

```sql
--  How can you find students who scored above 90 OR have perfect attendance?
    SELECT s.student_id,
    s.name,
    MAX(g.marks_obtained) AS highest_marks,
    ROUND((SUM(a.status = 'Present') * 100.0) / COUNT(a.attendance_id), 2)
        AS attendance_percentage
        FROM Students s
        LEFT JOIN Grades g
        ON s.student_id = g.student_id
        LEFT JOIN Attendance a
        ON s.student_id = a.student_id
        GROUP BY s.student_id, s.name
        HAVING highest_marks > 90
        OR attendance_percentage = 100;
```

student_id	name	highest_marks	attendance_percentage
101	Amit Sharma	95	100.00
102	Neha Patel	78	100.00
104	Priya Singh	88	100.00
106	Sneha Joshi	91	100.00

```sql
-- How can you list faculty members who are NOT assigned to any course?
SELECT
f.faculty_id,
f.name,
f.email
FROM Faculty f
LEFT JOIN Courses c
ON f.faculty_id = c.faculty_id
WHERE c.course_id IS NULL;
```

faculty_id	name	email
204	Prof Khan	khan@college.com

```sql
-- 4. Sorting & Grouping Data (ORDER BY, GROUP BY) (Medium Weightage)

-- How can you list students alphabetically by name?
select * from students order by name ;
```

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
101	Amit Sharma	2002-05-10	Male	amit@gmail.com	8878943218	Ahmedabad	2021-07-10	1
105	Karan Mehta	2002-11-25	Male	NULL	9876543214	Ahmedabad	2021-09-12	2
109	Kunal Sharma	2003-12-13	Male	kunal@gmail.com	9789897867	Ahmedabad	2021-08-12	1
102	Neha Patel	2003-02-15	Female	NULL	9876543211	Surat	2022-06-15	1
104	Priya Singh	2004-01-05	Female	priya@gmail.com	9876543213	Rajkot	2023-01-20	3
103	Rahul Verma	2001-09-20	Male	rahul@gmail.com	9876543212	Vadodara	2020-08-01	2
106	Sneha Joshi	2000-03-18	Female	sneha@gmail.com	9876543215	Pune	2019-07-05	1
107	Vikas Yadav	2003-06-30	Male	vikas@gmail.com	9876543216	Delhi	2022-02-11	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- How can you count the number of students enrolled in each department?
select department_id,count(*) from students group by department_id;
```

	department_id	count(*)
▶	1	4
	2	2
	3	1
	4	1

```sql
-- How can you show the average marks per course?
SELECT c.course_id, c.course_name,
AVG(g.marks_obtained) AS average_marks
FROM Grades g
JOIN Courses c
ON g.course_id = c.course_id
GROUP BY c.course_id, c.course_name;
```

	course_id	course_name	average_marks
▶	301	Database Systems	86.5000
	302	Operating Systems	91.0000
	303	Data Structures	56.0000
	304	Digital Electronics	88.0000
	305	Thermodynamics	35.0000

```sql
-- 5. Use Aggregate Functions (SUM, AVG, MAX, MIN, COUNT) (High Weightage)

-- How can you find the average attendance percentage of students?
SELECT
ROUND(AVG(attendance_percentage), 2) AS average_attendance_percentage
FROM (
    SELECT
    student_id,
    (SUM(status = 'Present') * 100.0) / COUNT(*)
        AS attendance_percentage
    FROM Attendance
    GROUP BY student_id
) AS student_attendance;
```

	average_attendance_percentage
▶	57.14

```sql
-- How can you identify the highest and lowest marks obtained in each course?
SELECT c.course_id,c.course_name,
MAX(g.marks_obtained) AS highest_marks,
MIN(g.marks_obtained) AS lowest_marks
FROM Grades g
JOIN Courses c
ON g.course_id = c.course_id
GROUP BY c.course_id, c.course_name;
```

course_id	course_name	highest_marks	lowest_marks
301	Database Systems	95	78
302	Operating Systems	91	91
303	Data Structures	67	45
304	Digital Electronics	88	88
305	Thermodynamics	35	35

```sql
-- How can you calculate the total number of students per department?
SELECT d.department_id,d.department_name,
COUNT(s.student_id) AS total_students
FROM Departments d
JOIN Students s
ON d.department_id = s.department_id
GROUP BY d.department_id, d.department_name;
```

department_id	department_name	total_students
1	Computer Science	4
2	Information Technology	2
3	Electronics	1
4	Mechanical	1

```sql
-- 7. Implement Joins
-- Retrieve student details along with their department (INNER JOIN)
SELECT s.student_id,s.name, s.email, s.phone_number, d.department_id, d.department_name
FROM Students s
INNER JOIN Departments d
ON s.department_id = d.department_id;
```

student_id	name	email	phone_number	department_id	department_name
101	Amit Sharma	amit@gmail.com	8878943218	1	Computer Science
102	Neha Patel	NULL	9876543211	1	Computer Science
106	Sneha Joshi	sneha@gmail.com	9876543215	1	Computer Science
109	Kunal Sharma	kunal@gmail.com	9789897867	1	Computer Science
103	Rahul Verma	rahul@gmail.com	9876543212	2	Information Technology
105	Karan Mehta	NULL	9876543214	2	Information Technology
104	Priya Singh	priya@gmail.com	9876543213	3	Electronics
107	Vikas Yadav	vikas@gmail.com	9876543216	4	Mechanical

```sql
-- Students who have not enrolled in any course (LEFT JOIN)
    SELECT s.student_id,s.name,s.email
    FROM Students s
    LEFT JOIN Enrollments e
    ON s.student_id = e.student_id
    WHERE e.enrollment_id IS NULL;
```

student_id	name	email

```sql
-- Courses that have no faculty assigned (RIGHT JOIN)
    SELECT c.course_id,c.course_name
    FROM Faculty f
    RIGHT JOIN Courses c
    ON f.faculty_id = c.faculty_id
    WHERE f.faculty_id IS NULL;
```

course_id	course_name
305	Thermodynamics

```sql
-- Show students without grades (FULL OUTER JOIN)
    SELECT s.student_id,s.name,g.marks_obtained
    FROM Students s
    LEFT JOIN Grades g
    ON s.student_id = g.student_id

    UNION

    SELECT s.student_id,s.name,g.marks_obtained
    FROM Students s
    RIGHT JOIN Grades g
    ON s.student_id = g.student_id;
```

student_id	name	marks_obtained
101	Amit Sharma	95
102	Neha Patel	78
103	Rahul Verma	45
104	Priya Singh	88
105	Karan Mehta	67
106	Sneha Joshi	91
107	Vikas Yadav	35
109	Kunal Sharma	NULL

```sql
-- 8. Use Subqueries (High Weightage)

--   How can you find students whose marks are above the average score?
    SELECT s.student_id, s.name,g.marks_obtained
    FROM Grades g
    JOIN Students s
    ON g.student_id = s.student_id
    WHERE g.marks_obtained > (
        SELECT AVG(marks_obtained)
        FROM Grades
    );
```

student_id	name	marks_obtained
101	Amit Sharma	95
102	Neha Patel	78
104	Priya Singh	88
106	Sneha Joshi	91

```sql
--   How can you identify students who have missed more than 10 classes?
    SELECT s.student_id,s.name,
    COUNT(*) AS missed_classes
    FROM Attendance a
    JOIN Students s
    ON a.student_id = s.student_id
    WHERE a.status = 'Absent'
    GROUP BY s.student_id, s.name
    HAVING COUNT(*) > 10;
```

	student_id	name	missed_classes

```sql
-- 9. Implement Date & Time Functions (High Weightage)

--   How can you extract the month from attendance_date to analyze attendance trends?
   select attendance_date, month(attendance_date) as Month_, monthname(attendance_date) as Month_name from attendance;
```

	attendance_date	Month_	Month_name
▶	2023-03-01	3	March
	2023-03-02	3	March
	2023-03-03	3	March
	2023-03-04	3	March
	2023-03-05	3	March
	2023-03-06	3	March
	2023-03-07	3	March

```sql
--  How can you calculate the number of years since a student's admission?
    select name,dob,admission_date,timestampdiff(year,dob,admission_date) as Num_of_Year From Students;
```

	name	dob	admission_date	Num_of_Year
▶	Amit Sharma	2002-05-10	2021-07-10	19
	Neha Patel	2003-02-15	2022-06-15	19
	Rahul Verma	2001-09-20	2020-08-01	18
	Priya Singh	2004-01-05	2023-01-20	19
	Karan Mehta	2002-11-25	2021-09-12	18
	Sneha Joshi	2000-03-18	2019-07-05	19
	Vikas Yadav	2003-06-30	2022-02-11	18
	Kunal Sharma	2003-12-13	2021-08-12	17

```
-- How can you format attendance_date as DD-MM-YYYY?
   select attendance_date, date_format(Attendance_date,'%d-%m-%Y') as format_date from Attendance;
```

	attendance_date	format_date
▶	2023-03-01	01-03-2023
	2023-03-02	02-03-2023
	2023-03-03	03-03-2023
	2023-03-04	04-03-2023
	2023-03-05	05-03-2023
	2023-03-06	06-03-2023
	2023-03-07	07-03-2023

```
-- 10. Use String Manipulation Functions (High Weightage)

-- How can you convert all faculty names to uppercase?
   select name,upper(name) from faculty;
```

	name	upper(name)
▶	Dr Mehta	DR MEHTA
	Prof Shah	PROF SHAH
	Dr Iyer	DR IYER
	Prof Khan	PROF KHAN
	Dr Patel	DR PATEL
	Dr Karan	DR KARAN

```
-- How can unnecessary spaces be trimmed from student names?
   select name,trim(name) from students;
```

	name	trim(name)
▶	Amit Sharma	Amit Sharma
	Neha Patel	Neha Patel
	Rahul Verma	Rahul Verma
	Priya Singh	Priya Singh
	Karan Mehta	Karan Mehta
	Sneha Joshi	Sneha Joshi
	Vikas Yadav	Vikas Yadav
	Kunal Sharma	Kunal Sharma

```sql
-- How can NULL email fields be replaced with "Email Not Provided"?
SELECT student_id,name,
IFNULL(email, 'Email Not Provided') AS Updated_email
FROM Students;
```

	student_id	name	Updated_email
▶	101	Amit Sharma	amit@gmail.com
	102	Neha Patel	Email Not Provided
	103	Rahul Verma	rahul@gmail.com
	104	Priya Singh	priya@gmail.com
	105	Karan Mehta	Email Not Provided
	106	Sneha Joshi	sneha@gmail.com
	107	Vikas Yadav	vikas@gmail.com
	109	Kunal Sharma	kunal@gmail.com

```sql
-- 11. Implement Window Functions (Very High Weightage)

-- How can students be ranked based on their overall marks?
SELECT
s.student_id,
s.name,
SUM(g.marks_obtained) AS total_marks,
RANK() OVER (ORDER BY SUM(g.marks_obtained) DESC) AS rank_position
FROM Grades g
JOIN Students s
ON g.student_id = s.student_id
GROUP BY s.student_id, s.name;
```

	student_id	name	total_marks	rank_position
▶	101	Amit Sharma	95	1
	106	Sneha Joshi	91	2
	104	Priya Singh	88	3
	102	Neha Patel	78	4
	105	Karan Mehta	67	5
	103	Rahul Verma	45	6
	107	Vikas Yadav	35	7

```sql
-- How can the cumulative attendance percentage per course be displayed?
SELECT course_id,
ROUND((SUM(status = 'Present') * 100.0) / COUNT(*), 2)
AS attendance_percentage
FROM Attendance
GROUP BY course_id;
```

	course_id	attendance_percentage
▶	301	100.00
	302	100.00
	303	0.00
	304	100.00
	305	0.00

```sql
-- How can the running total of students enrolled per month be shown?
SELECT month, monthly_enrollments,
SUM(monthly_enrollments)
    OVER (ORDER BY month) AS running_total
    FROM (
    SELECT
    DATE_FORMAT(enrollment_date, '%Y-%m') AS month,
    COUNT(*) AS monthly_enrollments
    FROM Enrollments
    GROUP BY DATE_FORMAT(enrollment_date, '%Y-%m')
) AS monthly_data;
```

	month	monthly_enrollments	running_total
▶	2023-01	3	3
	2023-02	3	6
	2023-03	3	9

```sql
-- 12. Apply SQL CASE Expressions (Very High Weightage)

--  How can student performance levels be assigned using conditions such as:

--      "Excellent" if marks obtained are greater than 90?
--      "Good" if marks obtained are between 75 and 90?
--      "Needs Improvement" otherwise?

        SELECT s.student_id,s.name,g.marks_obtained,
        CASE
            WHEN g.marks_obtained > 90 THEN 'Excellent'
            WHEN g.marks_obtained BETWEEN 75 AND 90 THEN 'Good'
            ELSE 'Needs Improvement'
        END AS performance_level
        FROM Grades g
        JOIN Students s
        ON g.student_id = s.student_id;
```

	student_id	name	marks_obtained	performance_level
▶	101	Amit Sharma	95	Excellent
	102	Neha Patel	78	Good
	103	Rahul Verma	45	Needs Improvement
	104	Priya Singh	88	Good
	105	Karan Mehta	67	Needs Improvement
	106	Sneha Joshi	91	Excellent
	107	Vikas Yadav	35	Needs Improvement

```sql
-- How can attendance records be categorized as:
--      "Regular" for attendance above 80%?
--      "Irregular" for attendance between 50% and 80%?
--      "defaulter" otherwise

        SELECT s.student_id,s.name,
        ROUND((SUM(a.status = 'Present') * 100.0) / COUNT(*), 2) AS attendance_percentage,
        CASE
            WHEN (SUM(a.status = 'Present') * 100.0) / COUNT(*) > 80 THEN 'Regular'
            WHEN (SUM(a.status = 'Present') * 100.0) / COUNT(*) BETWEEN 50 AND 80 THEN 'Irregular'
            ELSE 'Defaulter'
        END AS attendance_status
        FROM Attendance a
        JOIN Students s
        ON a.student_id = s.student_id
        GROUP BY s.student_id, s.name;
```

student_id	name	attendance_percentage	attendance_status
101	Amit Sharma	100.00	Regular
102	Neha Patel	100.00	Regular
103	Rahul Verma	0.00	Defaulter
104	Priya Singh	100.00	Regular
105	Karan Mehta	0.00	Defaulter
106	Sneha Joshi	100.00	Regular
107	Vikas Yadav	0.00	Defaulter