Institute Of Management System

Project for SQL Module by
Anuj Singh

1. Description:

Following database schema is designed to function as a backend storage database for a web

application built to manage a Institute.

By storing information in a relational database, all the tasks related to daily functioning of the

Institute can be performed easily and much more efficiently. Some of the benefits of using this system toerror

? RDBMS provides many ways to analyze available data, thus helping in making more informed

decisions about inventory management and other aspects of Institute management

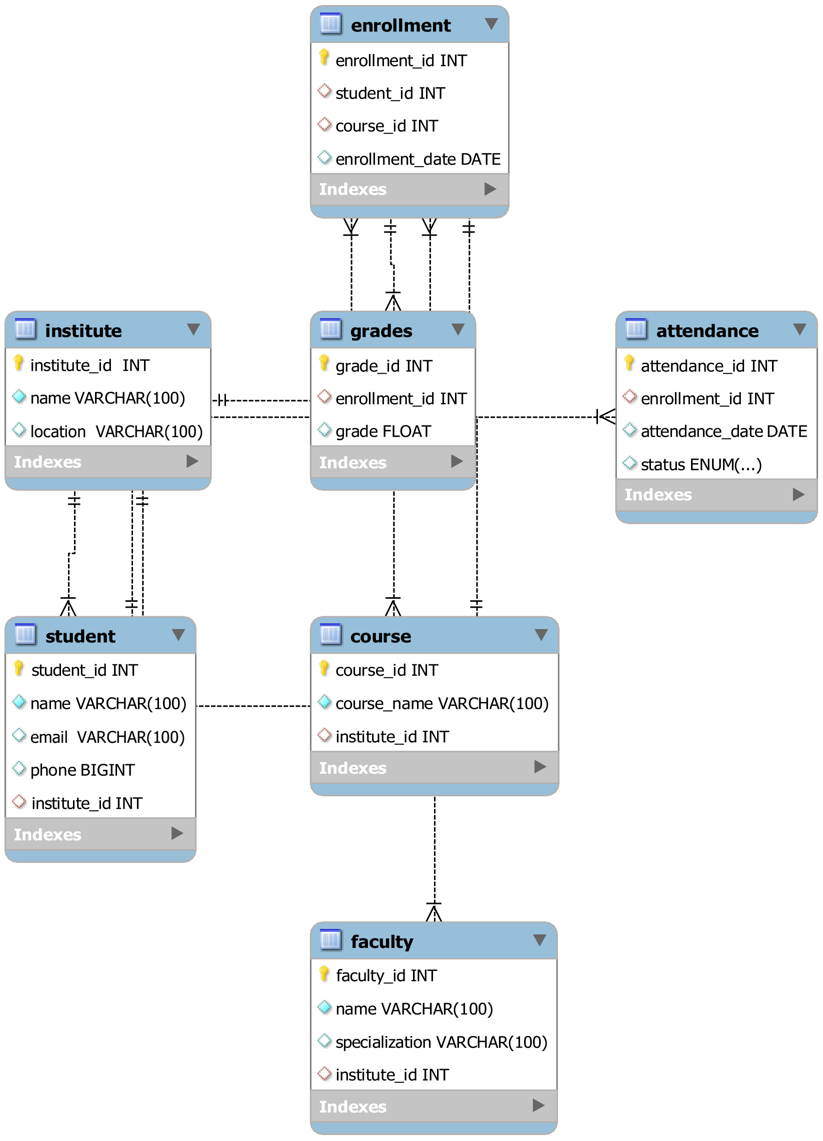
This database contains 7 tables:

1. Institute
2. Faculty
3. Course
4. Student
5. Attendance
6. Enrollment
7. Grades

How these tables/entities are related to each other is shown pictorially on next page through

ER diagram, i.e., Entity Relationship Diagram.

2. ER-Diagram for Institute Of Management System



1) CREATE DATABASE institute_of_management_system;USE institute_of_management_system;

```
-- Table: Institute

CREATE TABLE Institute (
   institute_id INT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   location VARCHAR(100)
);
desc Institute;
insert into Institute values(1,'itvedant','mumbai');

select * from Institute;

-- Table: Course
CREATE TABLE Course (
   course_id INT PRIMARY KEY,
   course_name VARCHAR(100) NOT NULL,
   institute_id INT,
   FOREIGN KEY (institute_id) REFERENCES Institute(institute_id)
);
desc course;
insert into Course values
('1','fs','1'),
('2','dse','1'),
('3','aws','1');
SELECT * FROM Course;
```

```
-- Table: Faculty

CREATE TABLE Faculty (
    faculty_id INT  PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    specialization VARCHAR(100),
    institute_id INT,
    FOREIGN KEY (institute_id) REFERENCES Institute(institute_id)
);
desc Faculty;
insert into Faculty values(1,'rohit','fs',1),(2,'modi','dse',1),(3,'yogi','aws',1);
select * from Faculty;


-- Table: Student
CREATE TABLE Student (
    student_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100),
    phone bigint,
    institute_id INT,
    FOREIGN KEY (institute_id) REFERENCES Institute(institute_id)
);
desc Student;
insert into Student values
(1,'anuj','ab@gmail.com','1234567','1'),
(2,'aniket','aab@gmail.com','123457','1'),
(3,'akash','cb@gmail.com','1234567','1'),
(4,'ankit','db@gmail.com','12345679','1'),
(5,'aniket','eab@gmail.com','1234570','1'),

(6,'akash','clb@gmail.com','12345687','1'),

(7,'rohit','d@gmail.com','120345679','1'),

(8,'virendra','eaab@gmail.com','12345970','1'),

(9,'salaman','cz@gmail.com','1234578687','1'),

(10,'shahrukh','edaggb@gmail.com','129934570','1'),

(11,'prabash','clkkb@gmail.com','1234560087','1'),

(12,'pawan','dhg@gmail.com','12034560079','1'),

(13,'krishna','ealab@gmail.com','1234500970','1'),

(14,'gagan','ckz@gmail.com','123450078687','1'),

(15,'rajiv','cksz@gmail.com','1450078687','1');

select * from student;



-- Table: Enrollment

CREATE TABLE Enrollment (

   enrollment_id INT PRIMARY KEY,

   student_id INT,

   course_id INT,

   enrollment_date DATE,

   FOREIGN KEY (student_id) REFERENCES Student(student_id),

   FOREIGN KEY (course_id) REFERENCES Course(course_id)

);

INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)

VALUES

(1,'1','1','2024-01-01'),

(2,'2','1','2024-01-01'),

(3,'3','1','2024-01-01'),
(4,'4','1','2024-01-01'),
(5,'5','1','2024-01-01'),
(6,'5','2','2024-02-01'),
(7,'7','2','2024-02-01'),
(8,'8','2','2024-02-01'),
(9,'9','2','2024-02-01'),
(10,'10','2','2024-02-01'),
(11,'11','3','2024-03-01'),
(12,'12','3','2024-03-01'),
(13,'13','3','2024-03-01'),
(14,'14','3','2024-03-01'),
(15,'15','3','2024-01-01');
desc enrollment;
select * from enrollment;

-- Table: Attendance
CREATE TABLE Attendance (
   attendance_id INT AUTO_INCREMENT PRIMARY KEY,
   enrollment_id INT,
   attendance_date DATE,
   status ENUM('Present', 'Absent'),
   FOREIGN KEY (enrollment_id) REFERENCES Enrollment(enrollment_id)
   );
   desc attendence;
INSERT INTO Attendance (enrollment_id, attendance_date, status)
VALUES
   (1, '2024-01-01', 'Present'),
   (2, '2024-01-01', 'Present'),
   (3, '2024-01-01', 'Present'),
    (4, '2024-01-01', 'Absent'),
    (5, '2024-01-01', 'Present'),
    (6, '2024-02-01', 'Absent'),
    (7, '2024-02-01', 'Present'),
    (8, '2024-02-01', 'Present'),
    (9, '2024-02-01', 'Absent'),
    (10, '2024-02-01', 'Present'),
    (11, '2024-03-01', 'Present'),
    (12, '2024-03-01', 'Present'),
    (13, '2024-03-01', 'Absent'),
    (14, '2024-03-01', 'Present'),
    (15, '2024-01-01', 'Present');

    select * from attendance;

-- Table: Grades
CREATE TABLE Grades (
    grade_id INT AUTO_INCREMENT PRIMARY KEY,
    enrollment_id INT,
    grade float,
    FOREIGN KEY (enrollment_id) REFERENCES Enrollment(enrollment_id)
);
INSERT INTO Grades (enrollment_id, grade)
VALUES
    (1, 85.5),
    (2, 90.0),
    (4, 60.8),
    (5, 95.7),
    (6, 70.3),
    (7, 88.9),

    (8, 79.4),

    (9, 82.1),

    (10, 91.6),

    (11, 85.2),

    (12, 78.5),

    (13, 64.7),

    (14, 93.0),

    (15, 87.3);

  desc Grades;

  select * from grades;

  SHOW TABLES;

select * from Institute inner join Course on Institute.Institute_id = Course.course_id;

  -- Query to retrieve all courses offered by an institute

  SELECT * FROM Course WHERE institute_id = 1;

  -- Query to find all students enrolled in a particular course

  SELECT s.name AS student_name, c.course_name

  FROM Student s

  JOIN Enrollment e ON s.student_id = e.student_id

  JOIN Course c ON e.course_id = c.course_id

  WHERE c.course_id = 3;

  -- Query to find faculty teaching a particular course
```

```sql
SELECT f.name AS faculty_name, c.course_name

FROM Faculty f

JOIN Course c ON f.institute_id = c.institute_id

WHERE c.course_id = 1;


-- Query to get attendance of a student for a particular course

SELECT a.attendance_date, a.status

FROM Attendance a

JOIN Enrollment e ON a.enrollment_id = e.enrollment_id

JOIN Student s ON e.student_id = s.student_id

JOIN Course c ON e.course_id = c.course_id

WHERE s.student_id = 7 AND c.course_id = 2;


-- Query to calculate average grade for a student in a particular course

SELECT AVG(grade) AS average_grade

FROM Grades

WHERE enrollment_id IN (SELECT enrollment_id FROM Enrollment WHERE student_id = 8 AND
course_id = 2);
```