

## **6. PROJECT: ONLINE COURSE MANAGEMENT SYSTEM**

### **ABSTRACT**

This project details the design and implementation of an Online Course Management System using Microsoft Access as the relational database engine. The core objective was to move administrative processes from inefficient spreadsheets to a structured database environment, specifically to manage the complex administrative load associated with Student Records, Course Offerings, and Enrollment Transactions. The system is engineered to support seamless data entry, rigorous data retrieval, and complex Financial and Statistical Reporting, thereby increasing operational efficiency and data reliability for the educational institution.

The technical foundation of the system is rooted in database normalization principles (adhering to at least 3NF) to eliminate data redundancy. This necessitated the resolution of the inherent Many-to-Many relationship between the Student and Course entities. This was achieved by introducing the dedicated Enrollments table, which acts as a Junction Table and enforces two One-to-Many relationships via Foreign Keys (StudentID and CourseID). The systematic use of Referential Integrity guarantees data validity, preventing illogical operations such as deleting an enrolled student or registering a student for a non-existent course.

The user interface emphasizes Data Integrity and Usability. Customized data entry Forms (Student Registration, Course Entry) utilize Combo Boxes on the Enrollment Form to force users to select human-readable names, while the underlying system correctly stores the hidden Primary Key identifiers. This functionality reduces data entry errors associated with manual ID input. Furthermore, several powerful Queries (e.g., Course Earnings Summary and Course Enrollments Count) leverage INNER JOINs and Aggregate Functions (SUM, COUNT) to produce critical analytical reports, effectively transforming transactional data into actionable business intelligence. The resulting application is a complete, robust, and user-centric solution ready for deployment and evaluation.

## SCREENSHOTS

### TABLES

#### 1. Student Table

	StudentID	Sname	Email	Contact	Click to Add
[+]	12	Jees	Jees@gmail.com	9828937276	
[+]	13	Lily	lily@gmail.com	8735678388	
[+]	14	Hari	hari@gmail.com	8373362890	
[+]	15	Mary	mary@gmail.com	8373563892	
[+]	16	Sally	sal@gmail.com	34216789420	
[+]	17	Diya George	diya@gmail.com	8765432122	
[+]	18	sajith	sajith@gmail.com	2178965443	
[+]	19	rena	rena@gmail.com	9656510521	
[+]	20	Merlin	merli@gmail.com	8765432104	
[+]	21	aadharsh	adharsh@gmai	234568976	
*	(New)				

The Student Table serves as a Master Data repository, storing all unique, non-repeating personal information for every individual student. Its primary purpose is to hold the fields essential for identification and contact, such as the student's name (Sname), email, and contact number. Crucially, this table is anchored by the AutoNumber Primary Key (StudentID), which guarantees that every student is assigned a unique identifier. This table functions as the "One" side of a One-to-Many relationship, meaning all enrollment records must ultimately link back to one specific student record contained here.

#### 2. Course Table

	CourseID	CourseName	Instructor	Fee	Click to Add
[+]	11	Data Structures	Paul	₹ 4,000.00	
[+]	12	Java	Shoby	₹ 5,000.00	
[+]	13	Android	Aleena	₹ 1,000.00	
[+]	14	Operating Systems	Dr Geetha	₹ 23,000.00	
[+]	15	Cyber Security	Dr Sathya	₹ 12,000.00	
[+]	16	Accountancy	Dr Charles	₹ 15,000.00	
[+]	17	Business Studies	Dr Benjamin	₹ 20,000.00	
*	(New)			₹ 0.00	

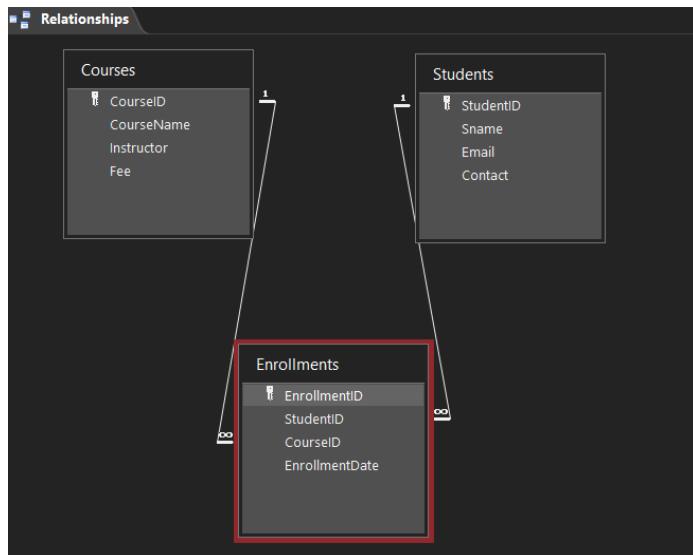
The Course Table is the second Master Data table, dedicated to storing details about every course offering. It holds descriptive and financial data, including the CourseName, the assigned Instructor, and the course Fee. Like the Student table, it is protected by its own AutoNumber Primary Key (CourseID), ensuring that each course title is uniquely catalogued. This table also acts as the "One" side of a One-to-Many relationship, ensuring that any transaction involving a course (like an enrollment or a revenue calculation) accurately references the course details stored in this central location.

### 3. Enrollments Table

EnrollmentID	StudentID	CourseID	EnrollmentDate	Click to Add
12	18	15	31-10-2025	
13	13	13	18-10-2025	
14	14	11	18-10-2025	
15	15	12	24-10-2025	
16	16	14	09-10-2025	
17	12	17	20-10-2025	
18	19	12	20-10-2025	
19	20	13	23-10-2025	
20	21	15	22-10-2025	
*	(New)	0	0	

The Enrollments Table is the operational heart of the system, functioning as the Junction Table that solves the Many-to-Many relationship between students and courses. It stores Transactional Data—details that are only relevant when a student and a course interact, such as the EnrollmentDay. The key feature of this table is that it contains two Foreign Keys (StudentID and CourseID). These foreign keys connect the table to the two master tables, allowing queries to trace exactly which course was taken by which student, thereby completing the robust, normalized data model.

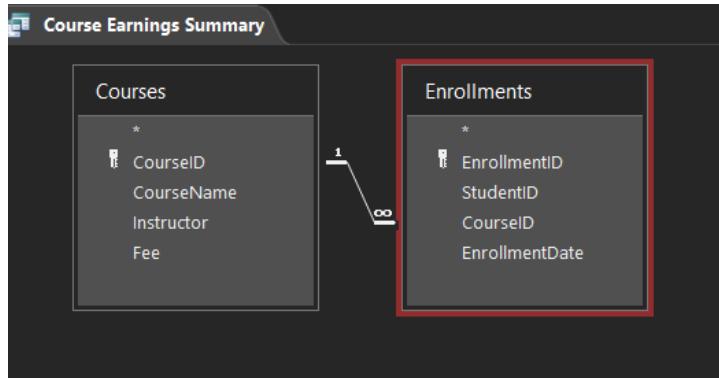
## RELATIONSHIP



This database system is structured to manage the Many-to-Many relationship between Students and Courses. This is solved by the Enrollments Table acting as a Junction Table. The system consists of two Master Tables (Students and Courses) which link to the central Enrollments Table through two separate One-to-Many relationships. This design uses Foreign Keys (StudentID and CourseID in the Enrollments table) to maintain Referential Integrity and accurately track who took what course and when.

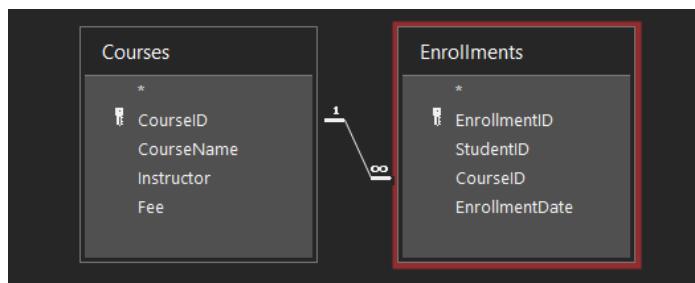
## QUERIES

### 1. Course Earnings Summary



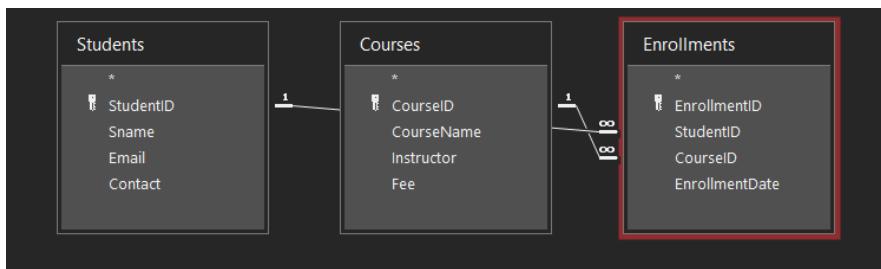
This query demonstrates a fundamental One-to-Many relationship usage for financial aggregation. It establishes a necessary INNER JOIN between the Course Table (the "One" side, which holds the unique Fee) and the Enrollments Table (the "Many" side, which tracks the number of times the fee should be counted). This ensures the correct fee is associated with every transaction. The query then uses the SUM aggregate function to tally these linked fees and applies a GROUP BY clause on CourseName, thereby presenting the total financial result for each specific course offering.

### 2. Course Enrollments Count



This query also leverages the One-to-Many link between the Course Table and the Enrollments Table to produce statistical results. By performing an INNER JOIN on the CourseID fields, the query effectively links all related transactions to their source course. The COUNT aggregate function is then applied to the resulting record set. Crucially, the GROUP BY CourseName clause is used to segment the counted transactions, ensuring the count is not a single total for the whole database, but rather an accurate measure of class size for each distinct course.

### 3. Student Enrollments Query



This query is a direct implementation of the Junction Table design used to resolve the Many-to-Many problem. It uses Nested INNER JOINS to chain all three tables together: first, linking the Course Table to the Enrollments Table, and second, linking the Student Table to that result. By selecting fields from all three entities (Student, Course, and Enrollment), the query provides a complete, denormalized record set that accurately reflects the full enrollment history, showing who took what course and when, based entirely on the validity of the foreign key links within the Enrollments table.

## FORMS

### 1. Course Entry Form

The form is titled "Courses". It contains four text input fields: "CourseID" with "(New)" placeholder, "CourseName", "Instructor", and "Fee" with "₹ 0.00" placeholder. Below the fields are two buttons: "Add Record" and "Save".

This form provides a straightforward interface for adding new course offerings to the Course Table. Its function is to capture the necessary details for each unique class, including the CourseName, the assigned Instructor, and the specific Fee. Like the Student Registration Form, it is designed for simple master data entry, ensuring that all course offerings are accurately catalogued before any student can enroll in them.

### 2. Enrollments Form

The form is titled "Enrollments". It contains five fields: "EnrollmentID" with "(New)" placeholder, "CourseID" with "0" placeholder, "EnrollmentDate", "Student" (dropdown menu), and "Course Title" (dropdown menu). Below the fields are two buttons: "Add Record" and "Save".

This is the most critical and technically sophisticated form, as it handles the transactional data that links your two master tables. Its defining feature is the use of Combo Boxes (dropdown menus) for the StudentID and CourseID fields. This is an essential data integrity tool: it allows the user to easily select the Student's name and Course Title, while the form correctly stores the underlying Primary Key numbers. This prevents errors and ensures that all enrollment records are validly linked across the database.

### **3. Student Registration Form**

The screenshot shows a user interface for a 'Students' form. At the top left is a small icon of a computer monitor. To its right, the word 'Students' is displayed in a light blue header bar. Below the header, there are four input fields: 'StudentID' with '(New)' placeholder text, 'Sname' (surname), 'Email', and 'Contact'. Each field has a corresponding label to its left. At the bottom of the form are two buttons: 'Add Record' (highlighted with a dashed border) and 'Save'.

This form is a simple, dedicated interface used for the initial input of new student records into the Student Table. Its primary role is to serve as a user-friendly screen for administrators to gather and save essential, non-repeating student details like the name, email, and contact information. It is crucial because every enrollment record relies on a student being accurately recorded here first.

## **REPORTS**

### **1. Course Earnings Summary**

Course Earnings Summary		24 October 2025 09:03:05
CourseName	Total Revenue Collected	
Android	₹ 2,000.00	
Business Studies	₹ 20,000.00	
Cyber Security	₹ 24,000.00	
Data Structures	₹ 4,000.00	
Java	₹ 10,000.00	
Operating System	₹ 23,000.00	
		83000

Page 1 of 1

This report is designed for financial analysis and administrative oversight. It is based on the query that uses the SUM aggregate function to calculate total revenue. The report's primary function is to display the total amount of fees collected for every single course offered. By presenting the data grouped by CourseName, it allows management to quickly identify which courses are the top revenue generators and track the financial performance of the department.

### **2. Course Enrollments Summary**

Course Enrollment Count		24 October 2025 08:12:36
CourseName	Total Enrollments	
Android	2	
Business Studies	1	
Cyber Security	2	
Data Structures	1	
Java	2	
Operating System	1	
	6	

Page 1 of 1

This report serves as the system's statistical and resource planning tool. It draws its data from the query that uses the COUNT aggregate function. The report's core purpose is to show the definitive total enrollment count (class size) for each course. This information is vital for scheduling classrooms, allocating instructors, and verifying the load of a specific offering before a new term begins.

### **3. Student Enrollments**

Student Enrollments			24 October 2025	09:04:34
Sname	CourseName	EnrollmentDate		
sajith	Cyber Security	31-10-2025		
Lily	Android	18-10-2025		
Hari	Data Structures	18-10-2025		
Mary	Java	24-10-2025		
Sally	Operating System	09-10-2025		
Jees	Business Studies	20-10-2025		
rena	Java	20-10-2025		
Merlin	Android	23-10-2025		
aadharsh	Cyber Security	22-10-2025		

9

Page 1 of 1

This report is a comprehensive record-keeping document detailing the full enrollment history of the students. It utilizes the results of the Master Join Query that links all three tables. The report lists every transaction, showing the Student Name, the Course Name, and the Enrollment Date. Because it is typically designed to be grouped by Student Name, it functions as the official student transcript, providing a clean, chronological history of courses taken by each individual.