INTRODUCTION

The University Management System (UMS) is a database-driven application designed to efficiently manage and organize the academic and administrative activities of a university. It serves as a centralized system to handle various aspects of university operations such as student records, faculty information, course registrations, departmental structures, hostel allocations, and examination details.

In a real-world university setting, managing large volumes of data manually can lead to inefficiencies, errors, and a lack of coordination between departments. The UMS addresses these challenges by offering a structured and relational database solution built using principles of Database Management Systems (DBMS). It ensures data integrity, reduces redundancy, and supports easy retrieval and manipulation of data.

This project implements several interconnected tables including **students**, **faculty**, **departments**, **courses**, **teaches**, **enrollments**, **hostels**, and **exams**, each with at least 20 sample entries. These tables are normalized and related using primary and foreign keys to maintain consistency and logical relationships among data entities.

By providing a reliable backend database design, this University Management System acts as a foundational tool for developing a complete university ERP system, paving the way for future enhancements like automation, reporting, and real-time data access.

OBJECTIVE

The primary objective of the University Management System project is to design and implement a relational database that efficiently manages the core academic and administrative data of a university.

This system aims to:

- Organize and store data related to students, faculty, departments, courses, hostels, and examinations.
- Establish proper relationships between different entities using primary and foreign keys.
- Ensure data consistency, integrity, and security through normalization and constraints.
- Provide easy and efficient data retrieval, insertion, updating, and deletion operations using SQL.
- Reduce redundancy and manual workload through a centralized and systematic data structure.
- Lay the foundation for further development of a complete university ERP system.

MySQL Language

MySQL is an open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL) to manage and manipulate databases. It is widely used in both academic and industrial applications due to its reliability, performance, and ease of use.

MySQL allows users to create, retrieve, update, and delete data efficiently while ensuring data integrity and security. It supports complex queries, transactions, indexing, and stored procedures, making it suitable for large-scale systems like a University Management System.

Some key features of MySQL include:

- Open-source and free under the GNU General Public License (GPL)
- Cross-platform compatibility
- Support for ACID compliance for reliable transaction processing
- Scalability and flexibility for managing growing datasets
- Integration with various programming languages and tools

In this project, MySQL is used as the backend database to manage the core functionalities of the University Management System through the design and implementation of normalized relational tables.

MySQL Code

```
-- University Management System - MySQL Code

-- 1. Departments Table
CREATE TABLE departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(100) NOT NULL
);

-- 2. Students Table
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    gender VARCHAR(10),
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

-- 3. Faculty Table
CREATE TABLE faculty (
    faculty_id INT PRIMARY KEY,
    name VARCHAR(100),
    dept_id INT,
    designation VARCHAR(50),
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

-- 4. Courses Table
CREATE TABLE courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(100),
    credits INT,
```

```
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

-- 5. Teaches Table
CREATE TABLE teaches (
    faculty_id INT,
    course_id INT,
    semester VARCHAR(10),
    PRIMARY KEY (faculty_id, course_id),
    FOREIGN KEY (faculty_id) REFERENCES faculty(faculty_id),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);

-- 6. Enrollments Table
CREATE TABLE enrollments (
    student_id INT,
    course_id INT,
    semester VARCHAR(10),
    grade VARCHAR(2),
    PRIMARY KEY (student_id, course_id),
    FOREIGN KEY (student_id) REFERENCES students(student_id),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);

-- 7. Hostel Table
CREATE TABLE hostel (
    hostel_id INT PRIMARY KEY,
    hostel_name VARCHAR(100),
    capacity INT
);

-- 8. Hostel Allocation
CREATE TABLE hostel_allocation (
    student_id INT PRIMARY KEY,
    hostel_id INT,
    room_number VARCHAR(10),
```

```
    FOREIGN KEY (student_id) REFERENCES students(student_id),
    FOREIGN KEY (hostel_id) REFERENCES hostel(hostel_id)
);

-- 9. Exams Table
CREATE TABLE exams (
    exam_id INT PRIMARY KEY,
    course_id INT,
    exam_date DATE,
    exam_type VARCHAR(20),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);

-- INSERT DATA FOR departments
INSERT INTO departments VALUES
(1, 'Computer Science'), (2, 'Electronics'), (3, 'Mechanical'), (4,
'Civil'), (5, 'Electrical'),
(6, 'Information Technology'), (7, 'Chemical'), (8, 'Biotechnology'), (9,
'Mathematics'), (10, 'Physics'),
(11, 'Chemistry'), (12, 'Management'), (13, 'Economics'), (14,
'English'), (15, 'History'),
(16, 'Geography'), (17, 'Political Science'), (18, 'Sociology'), (19,
'Philosophy'), (20, 'Statistics');

-- INSERT DATA FOR students
INSERT INTO students VALUES
(1, 'Alice', 20, 'Female', 1), (2, 'Bob', 21, 'Male', 2), (3, 'Charlie', 22,
'Male', 3), (4, 'Daisy', 19, 'Female', 4),
(5, 'Ethan', 20, 'Male', 5), (6, 'Fiona', 21, 'Female', 6), (7, 'George', 22,
'Male', 7), (8, 'Hannah', 19, 'Female', 8),
(9, 'Ian', 20, 'Male', 9), (10, 'Jane', 21, 'Female', 10), (11, 'Kyle', 22,
'Male', 11), (12, 'Lily', 19, 'Female', 12),
(13, 'Mike', 20, 'Male', 13), (14, 'Nina', 21, 'Female', 14), (15, 'Oscar',
22, 'Male', 15), (16, 'Paula', 19, 'Female', 16),
(17, 'Quinn', 20, 'Male', 17), (18, 'Rachel', 21, 'Female', 18), (19,
'Sam', 22, 'Male', 19), (20, 'Tina', 19, 'Female', 20);
```

```
-- INSERT DATA FOR faculty
INSERT INTO faculty VALUES
(1, 'Dr. Smith', 1, 'Professor'), (2, 'Dr. Clark', 2, 'Associate Professor'),
(3, 'Dr. Adams', 3, 'Professor'),
(4, 'Dr. Baker', 4, 'Assistant Professor'), (5, 'Dr. Jones', 5, 'Professor'),
(6, 'Dr. Lee', 6, 'Associate Professor'),
(7, 'Dr. White', 7, 'Professor'), (8, 'Dr. Green', 8, 'Assistant Professor'),
(9, 'Dr. Black', 9, 'Professor'),
(10, 'Dr. Brown', 10, 'Associate Professor'), (11, 'Dr. Taylor', 11,
'Professor'), (12, 'Dr. Hall', 12, 'Assistant Professor'),
(13, 'Dr. King', 13, 'Professor'), (14, 'Dr. Wright', 14, 'Associate
Professor'), (15, 'Dr. Lopez', 15, 'Professor'),
(16, 'Dr. Hill', 16, 'Assistant Professor'), (17, 'Dr. Scott', 17,
'Professor'), (18, 'Dr. Young', 18, 'Associate Professor'),
(19, 'Dr. Walker', 19, 'Professor'), (20, 'Dr. Allen', 20, 'Assistant
Professor');

-- INSERT DATA FOR courses
INSERT INTO courses VALUES
(1, 'Data Structures', 4, 1), (2, 'Digital Electronics', 3, 2), (3,
'Thermodynamics', 3, 3), (4, 'Structural Engineering', 4, 4),
(5, 'Power Systems', 3, 5), (6, 'Networking', 3, 6), (7, 'Fluid
Mechanics', 4, 7), (8, 'Genetics', 3, 8),
(9, 'Algebra', 3, 9), (10, 'Quantum Physics', 4, 10), (11, 'Organic
Chemistry', 4, 11), (12, 'Marketing', 3, 12),
(13, 'Microeconomics', 3, 13), (14, 'Literature', 2, 14), (15, 'Modern
History', 3, 15), (16, 'Cartography', 2, 16),
(17, 'Political Theory', 3, 17), (18, 'Social Research', 3, 18), (19,
'Logic', 2, 19), (20, 'Probability', 3, 20);

-- INSERT DATA FOR teaches
INSERT INTO teaches VALUES
(1, 1, 'S1'), (2, 2, 'S1'), (3, 3, 'S1'), (4, 4, 'S1'), (5, 5, 'S1'),
(6, 6, 'S1'), (7, 7, 'S1'), (8, 8, 'S1'), (9, 9, 'S1'), (10, 10, 'S1'),
(11, 11, 'S1'), (12, 12, 'S1'), (13, 13, 'S1'), (14, 14, 'S1'), (15, 15, 'S1'),
(16, 16, 'S1'), (17, 17, 'S1'), (18, 18, 'S1'), (19, 19, 'S1'), (20, 20, 'S1');
```

```
-- INSERT DATA FOR enrollments
INSERT INTO enrollments VALUES
(1, 1, 'S1', 'A'), (2, 2, 'S1', 'B'), (3, 3, 'S1', 'C'), (4, 4, 'S1', 'A'), (5, 5, 'S1', 'B'),
(6, 6, 'S1', 'A'), (7, 7, 'S1', 'C'), (8, 8, 'S1', 'B'), (9, 9, 'S1', 'A'), (10, 10, 'S1', 'A'),
(11, 11, 'S1', 'B'), (12, 12, 'S1', 'C'), (13, 13, 'S1', 'A'), (14, 14, 'S1', 'B'), (15, 15, 'S1', 'A'),
(16, 16, 'S1', 'C'), (17, 17, 'S1', 'A'), (18, 18, 'S1', 'B'), (19, 19, 'S1', 'C'), (20, 20, 'S1', 'A');

-- INSERT DATA FOR hostel
INSERT INTO hostel VALUES
(1, 'Alpha', 100), (2, 'Beta', 90), (3, 'Gamma', 80), (4, 'Delta', 70), (5, 'Epsilon', 60),
(6, 'Zeta', 50), (7, 'Eta', 45), (8, 'Theta', 40), (9, 'Iota', 35), (10, 'Kappa', 30),
(11, 'Lambda', 25), (12, 'Mu', 20), (13, 'Nu', 15), (14, 'Xi', 10), (15, 'Omicron', 8),
(16, 'Pi', 6), (17, 'Rho', 4), (18, 'Sigma', 3), (19, 'Tau', 2), (20, 'Upsilon', 1);

-- INSERT DATA FOR hostel_allocation
INSERT INTO hostel_allocation VALUES
(1, 1, 'A101'), (2, 2, 'B102'), (3, 3, 'C103'), (4, 4, 'D104'), (5, 5, 'E105'),
(6, 6, 'F106'), (7, 7, 'G107'), (8, 8, 'H108'), (9, 9, 'I109'), (10, 10, 'J110'),
(11, 11, 'K111'), (12, 12, 'L112'), (13, 13, 'M113'), (14, 14, 'N114'), (15, 15, 'O115'),
(16, 16, 'P116'), (17, 17, 'Q117'), (18, 18, 'R118'), (19, 19, 'S119'), (20, 20, 'T120');

-- INSERT DATA FOR exams
INSERT INTO exams VALUES
(1, 1, '2025-04-01', 'Midterm'), (2, 2, '2025-04-02', 'Midterm'), (3, 3, '2025-04-03', 'Midterm'),
```

(4, 4, '2025-04-04', 'Midterm'), (5, 5, '2025-04-05', 'Midterm'), (6, 6, '2025-04-06', 'Midterm'),
(7, 7, '2025-04-07', 'Midterm'), (8, 8, '2025-04-08', 'Midterm'), (9, 9, '2025-04-09', 'Midterm'),
(10, 10, '2025-04-10', 'Midterm'), (11, 11, '2025-04-11', 'Midterm'), (12, 12, '2025-04-12', 'Midterm'),
(13, 13, '2025-04-13', 'Midterm'), (14, 14, '2025-04-14', 'Midterm'), (15, 15, '2025-04-15', 'Midterm'),
(16, 16, '2025-04-16', 'Midterm'), (17, 17, '2025-04-17', 'Midterm'), (18, 18, '2025-04-18', 'Midterm'),
(19, 19, '2025-04-19', 'Midterm'), (20, 20, '2025-04-20', 'Midterm');

-- USEFUL QUERIES

-- 1. Table Name: departments
SELECT * FROM departments;

-- 2. Table Name: students
SELECT * FROM students;

-- 3. Table Name: faculty
SELECT * FROM faculty;

-- 4. Table Name: courses
SELECT * FROM courses;

-- 5. Table Name: teaches
SELECT * FROM teaches;

-- 6. Table Name: enrollments
SELECT * FROM enrollments;

-- 7. Table Name: hostel
SELECT * FROM hostel;

-- 8. Table Name: hostel_allocation
SELECT * FROM hostel_allocation;

```
-- 9. Table Name: exams
SELECT * FROM exams;

-- 10. List all students in a specific department (e.g., Computer
Science)
SELECT s.student_id, s.name, d.dept_name
FROM students s
JOIN departments d ON s.dept_id = d.dept_id
WHERE d.dept_name = 'Computer Science';

-- 11. View courses taught by a specific faculty (e.g., Dr. Smith)
SELECT f.name AS faculty_name, c.course_name, t.semester
FROM faculty f
JOIN teaches t ON f.faculty_id = t.faculty_id
JOIN courses c ON t.course_id = c.course_id
WHERE f.name = 'Dr. Smith';

-- 12. See all enrollments of a student (e.g., Alice)
SELECT s.name AS student_name, c.course_name, e.semester,
e.grade
FROM students s
JOIN enrollments e ON s.student_id = e.student_id
JOIN courses c ON e.course_id = c.course_id
WHERE s.name = 'Alice';

-- 13. Find faculty in a specific department (e.g., Electronics)
SELECT f.name, f.designation, d.dept_name
FROM faculty f
JOIN departments d ON f.dept_id = d.dept_id
WHERE d.dept_name = 'Electronics';

-- 14. List exams scheduled for a particular course (e.g., Data
Structures)
SELECT c.course_name, e.exam_date, e.exam_type
FROM exams e
JOIN courses c ON e.course_id = c.course_id
WHERE c.course_name = 'Data Structures';
```

```
-- 15. Show hostel allocations
SELECT s.name AS student_name, h.hostel_name, ha.room_number
FROM students s
JOIN hostel_allocation ha ON s.student_id = ha.student_id
JOIN hostel h ON ha.hostel_id = h.hostel_id;

-- 16. Get all students along with their department and hostel
SELECT s.student_id, s.name, d.dept_name, h.hostel_name
FROM students s
JOIN departments d ON s.dept_id = d.dept_id
LEFT JOIN hostel_allocation ha ON s.student_id = ha.student_id
LEFT JOIN hostel h ON ha.hostel_id = h.hostel_id;

-- 17. Courses offered by each department
SELECT d.dept_name, c.course_name, c.credits
FROM departments d
JOIN courses c ON d.dept_id = c.dept_id
ORDER BY d.dept_name;
```

SCREENSHOT(OUTPUT)

1st Output: Departments Table

dept_id	dept_name
1	Computer Science
2	Electronics
3	Mechanical
4	Civil
5	Electrical
6	Information Technology
7	Chemical
8	Biotechnology
9	Mathematics
10	Physics
11	Chemistry
12	Management
13	Economics
14	English
15	History
16	Geography
17	Political Science
18	Sociology
19	Philosophy
20	Statistics

2nd Output: Students Table

student_id	name	age	gender	dept_id
1	Alice	20	Female	1
2	Bob	21	Male	2
3	Charlie	22	Male	3
4	Daisy	19	Female	4
5	Ethan	20	Male	5
6	Fiona	21	Female	6
7	George	22	Male	7
8	Hannah	19	Female	8
9	Ian	20	Male	9
10	Jane	21	Female	10
11	Kyle	22	Male	11
12	Lily	19	Female	12
13	Mike	20	Male	13
14	Nina	21	Female	14
15	Oscar	22	Male	15
16	Paula	19	Female	16
17	Quinn	20	Male	17
18	Rachel	21	Female	18
19	Sam	22	Male	19
20	Tina	19	Female	20

3rd Output: Faculty Table

faculty_id	name	dept_id	designation
1	Dr. Smith	1	Professor
2	Dr. Clark	2	Associate Professor
3	Dr. Adams	3	Professor
4	Dr. Baker	4	Assistant Professor
5	Dr. Jones	5	Professor
6	Dr. Lee	6	Associate Professor
7	Dr. White	7	Professor
8	Dr. Green	8	Assistant Professor
9	Dr. Black	9	Professor
10	Dr. Brown	10	Associate Professor
11	Dr. Taylor	11	Professor
12	Dr. Hall	12	Assistant Professor
13	Dr. King	13	Professor
14	Dr. Wright	14	Associate Professor
15	Dr. Lopez	15	Professor
16	Dr. Hill	16	Assistant Professor
17	Dr. Scott	17	Professor
18	Dr. Young	18	Associate Professor
19	Dr. Walker	19	Professor
20	Dr. Allen	20	Assistant Professor

4th Output: Courses Table

course_id	course_name	credits	dept_id
1	Data Structures	4	1
2	Digital Electronics	3	2
3	Thermodynamics	3	3
4	Structural Engineering	4	4
5	Power Systems	3	5
6	Networking	3	6
7	Fluid Mechanics	4	7
8	Genetics	3	8
9	Algebra	3	9
10	Quantum Physics	4	10
11	Organic Chemistry	4	11
12	Marketing	3	12
13	Microeconomics	3	13
14	Literature	2	14
15	Modern History	3	15
16	Cartography	2	16
17	Political Theory	3	17
18	Social Research	3	18
19	Logic	2	19
20	Probability	3	20

5th Output: Teaches Table

faculty_id	course_id	semester
1	1	S1
2	2	S1
3	3	S1
4	4	S1
5	5	S1
6	6	S1
7	7	S1
8	8	S1
9	9	S1
10	10	S1
11	11	S1
12	12	S1
13	13	S1
14	14	S1
15	15	S1
16	16	S1
17	17	S1
18	18	S1
19	19	S1
20	20	S1

6th Output: Enrollments Table

student_id	course_id	semester	grade
1	1	S1	A
2	2	S1	B
3	3	S1	C
4	4	S1	A
5	5	S1	B
6	6	S1	A
7	7	S1	C
8	8	S1	B
9	9	S1	A
10	10	S1	A
11	11	S1	B
12	12	S1	C
13	13	S1	A
14	14	S1	B
15	15	S1	A
16	16	S1	C
17	17	S1	A
18	18	S1	B
19	19	S1	C
20	20	S1	A

7th Output: Hostel Table

hostel_id	hostel_name	capacity
1	Alpha	100
2	Beta	90
3	Gamma	80
4	Delta	70
5	Epsilon	60
6	Zeta	50
7	Eta	45
8	Theta	40
9	Iota	35
10	Kappa	30
11	Lambda	25
12	Mu	20
13	Nu	15
14	Xi	10
15	Omicron	8
16	Pi	6
17	Rho	4
18	Sigma	3
19	Tau	2
20	Upsilon	1

8th Output: Hostel_allocation Table

student_id	hostel_id	room_number
1	1	A101
2	2	B102
3	3	C103
4	4	D104
5	5	E105
6	6	F106
7	7	G107
8	8	H108
9	9	I109
10	10	J110
11	11	K111
12	12	L112
13	13	M113
14	14	N114
15	15	O115
16	16	P116
17	17	Q117
18	18	R118
19	19	S119
20	20	T120

9th Output: Exams Table

exam_id	course_id	exam_date	exam_type
1	1	2025-04-01	Midterm
2	2	2025-04-02	Midterm
3	3	2025-04-03	Midterm
4	4	2025-04-04	Midterm
5	5	2025-04-05	Midterm
6	6	2025-04-06	Midterm
7	7	2025-04-07	Midterm
8	8	2025-04-08	Midterm
9	9	2025-04-09	Midterm
10	10	2025-04-10	Midterm
11	11	2025-04-11	Midterm
12	12	2025-04-12	Midterm
13	13	2025-04-13	Midterm
14	14	2025-04-14	Midterm
15	15	2025-04-15	Midterm
16	16	2025-04-16	Midterm
17	17	2025-04-17	Midterm
18	18	2025-04-18	Midterm
19	19	2025-04-19	Midterm
20	20	2025-04-20	Midterm

10th Output: List all students in a specific department (e.g., Computer Science)

student_id	name	dept_name
1	Alice	Computer Science

11th Output: View courses taught by a specific faculty (e.g., Dr. Smith)

faculty_name	course_name	semester
Dr. Smith	Data Structures	S1

12th Output: See all enrollments of a student (e.g., Alice)

student_name	course_name	semester	grade
Alice	Data Structures	S1	A

13th Output: Find faculty in a specific department (e.g., Electronics)

name	designation	dept_name
Dr. Clark	Associate Professor	Electronics

14th Output: List exams scheduled for a particular course (e.g., Data Structures)

course_name	exam_date	exam_type
Data Structures	2025-04-01	Midterm

15th Output: Show hostel allocations

student_name	hostel_name	room_number
Alice	Alpha	A101
Bob	Beta	B102
Charlie	Gamma	C103
Daisy	Delta	D104
Ethan	Epsilon	E105
Fiona	Zeta	F106
George	Eta	G107
Hannah	Theta	H108
Ian	Iota	I109
Jane	Kappa	J110
Kyle	Lambda	K111
Lily	Mu	L112
Mike	Nu	M113
Nina	Xi	N114
Oscar	Omicron	O115
Paula	Pi	P116
Quinn	Rho	Q117
Rachel	Sigma	R118
Sam	Tau	S119
Tina	Upsilon	T120

16th Output: Get all students along with their department and hostel

student_id	name	dept_name	hostel_name
1	Alice	Computer Science	Alpha
2	Bob	Electronics	Beta
3	Charlie	Mechanical	Gamma
4	Daisy	Civil	Delta
5	Ethan	Electrical	Epsilon
6	Fiona	Information Technology	Zeta
7	George	Chemical	Eta
8	Hannah	Biotechnology	Theta
9	Ian	Mathematics	Iota
10	Jane	Physics	Kappa
11	Kyle	Chemistry	Lambda
12	Lily	Management	Mu
13	Mike	Economics	Nu
14	Nina	English	Xi
15	Oscar	History	Omicron
16	Paula	Geography	Pi
17	Quinn	Political Science	Rho
18	Rachel	Sociology	Sigma
19	Sam	Philosophy	Tau
20	Tina	Statistics	Upsilon

17th Output: Courses offered by each department

dept_name	course_name	credits
Biotechnology	Genetics	3
Chemical	Fluid Mechanics	4
Chemistry	Organic Chemistry	4
Civil	Structural Engineering	4
Computer Science	Data Structures	4
Economics	Microeconomics	3
Electrical	Power Systems	3
Electronics	Digital Electronics	3
English	Literature	2
Geography	Cartography	2
History	Modern History	3
Information Technology	Networking	3
Management	Marketing	3
Mathematics	Algebra	3
Mechanical	Thermodynamics	3
Philosophy	Logic	2
Physics	Quantum Physics	4
Political Science	Political Theory	3
Sociology	Social Research	3
Statistics	Probability	3

CONCLUSION

The University Management System (UMS) project successfully demonstrates the effective use of a relational database to manage and organize the essential operations of a university. By implementing this system using MySQL, we have created a structured and efficient way to store and retrieve information related to students, faculty, departments, courses, hostels, and examinations.

Through the use of normalization, relational integrity, and SQL queries, the project ensures minimal data redundancy, consistent relationships between entities, and accurate data handling. This system not only simplifies administrative processes but also provides a strong foundation for future expansion, such as adding a user interface, automation features, and advanced reporting tools.

Overall, this project highlights the practical importance of Database Management Systems in solving real-world problems and showcases how database design can improve the efficiency and reliability of institutional data management.