

Task 12

DBMS Project Summary - Online Learning Platform Management System (OLPMS)

1. Introduction:

Education has undergone a major transformation due to advancements in digital technology. Traditional classroom learning, while structured and interactive, suffers from limitations like geographical barriers, rigid schedules, and administrative inefficiencies. The Online Learning Platform Management System (OLPMS) is designed to address these issues by providing a centralized, digital environment for students, instructors, and administrators.

The system focuses on three main benefits:

- Flexibility and accessibility for students and teachers
- Administrative efficiency through automation and reduced errors.
- Data security and scalability ensuring reliable management of sensitive academic data

2. Problem Statement

Traditional classroom systems face several limitations:

- Limited accessibility: Students in remote areas find it difficult to attend physical classes

- Rigid schedules: Fixed timings restrict flexibility for working learners

- Manual processes: Administrative processes like enrollment, track are time-consuming and error-prone.

- Data Security system: Unstructured and poorly managed data like breaches

3. Objectives

The primary goals of the project are:

1. To design a relational database using Oracle for managing online learning data.
2. To ensure data integrity through normalization and relational constraints.
3. To integrate MongoDB for multimedia and unstructured data storage.
4. To demonstrate CRUD operations (Create, Read, Update, Delete) using SQL and NoSQL.
5. To ensure scalability, flexibility and efficient data management for academic institutions.

4. System Requirements

Hardware:

• Processor: Intel i5 or higher

• RAM: 8 GB minimum

• Storage: 250 GB HDD/SSD

Software:

• OS: Windows 10/Linux

• Database: Oracle 12c or higher

• Front-end: Java/Web interface

• NoSQL: MongoDB 6.0

• Tools: Oracle SQL Developer, MongoDB Compass

5. System Analysis and Design

Key Entities:

- Student
- Instructor
- Course

- Enrollment
- Assignment
- Submission
- Grade.

Relationships:

- Student \rightarrow Course (many to many)
- Course \rightarrow Assignment (one to many)
- Assignment \rightarrow Submission (one to many)
- Submission \rightarrow Grade (one to one)
- Instructor \rightarrow Course (one to many)

6. Schema Design

Example SQL

```
SQL > CREATE TABLE student (
    Student_ID NUMBER PRIMARY KEY,
    Student_Name VARCHAR2(50),
    Email VARCHAR2(100),
    Department VARCHAR2(30)
);

CREATE TABLE Instructor (
    Instructor_ID NUMBER PRIMARY KEY,
    Instructor_Name VARCHAR2(50),
    Email VARCHAR2(100),
    Department VARCHAR2(30)
);

CREATE TABLE LOUXE (
    Course_ID NUMBER PRIMARY KEY,
    Course_Name VARCHAR2(50),
    Credits NUMBER,
    Instructor_ID NUMBER REFERENCES
    Instructor (Instructor_ID)
);
```

4. Normalization

The design adheres to Third Normal form (3NF):

- 1NF: Each field holds atomic values

- 2NF: Non-key attributes depend fully on the primary key.

- 3NF: No transitive dependencies exist.

This ensures minimal redundancy, maximum integrity and efficient data retrieval

5. Implementation (SQL Queries)

Sample Data Insertion:

```
SQL> INSERT INTO student VALUES (101, 'John Doe',  
    'joh@gmail.com', 'CSE');
```

```
INSERT INTO Instr student VALUES (201, 'Dr. Smith',  
    'smith@edu.com', 'CSE');
```

```
INSERT INTO course VALUES (201, 'Database Systems',  
    4, 201);
```

```
INSERT INTO enrollment VALUES (101, 201, SYSDATE);
```

Sample Query:

```
SQL> SELECT s.Student_Name, c.Course_Name
```

```
FROM student s
```

```
JOIN enrollment e ON e.Student_ID =
```

```
e_Student_ID
```

```
JOIN course c ON e.Course_ID = c.Course_ID;
```

This retrieves all courses enrolled by each student

9. Integration with MongoDB (NoSQL)

To handle multimedia and unstructured data MongoDB complements the Oracle Database.

Database: Online Learning DB

Example collections:

Students - Students details and enrolled courses

Courses - Course materials

Assignments - Assignment details.

Submissions - Student submissions and feedback.

Sample MongoDB Query:

```
db.students.insertOne({
  student_id: "STU101",
  name: "Arun Kumar",
  email: "arun@example.com",
  department: "IT",
  enrolled_courses: ["C101"]
});
```

```
db.submissions.find({marks: {$gte: 40}},
  {student_id: 1, marks: 1, feedback: 1, _id: 0});
```

The hybrid approach combines structured relational data (Oracle) and unstructured media data (MongoDB) enhancing scalability and performance

Conclusion:

The Online Learning Platform Management System effectively digitizes education management. It combines Oracle's relational capabilities with MongoDB's NoSQL flexibility to build a secure, efficient and scalable environment. The system overcomes the limitations of traditional learning

by offering continuous access to resources, real-time tracking and automated administration

11. References:

1. Oracle database documentation - Oracle corporation

2. MongoDB official documentation

3. Silberschatz, Korth & Sudarshan - Database System Concepts

4. Raghu Ramakrishnan - Database Management Systems.

Result:- The hotel reservation management system is fully functional, tested and ready for deployment with all deliverables meeting the specified requirements