DBMS LAB

Name – RAVI SHEKHAR

Roll – 22CS3075

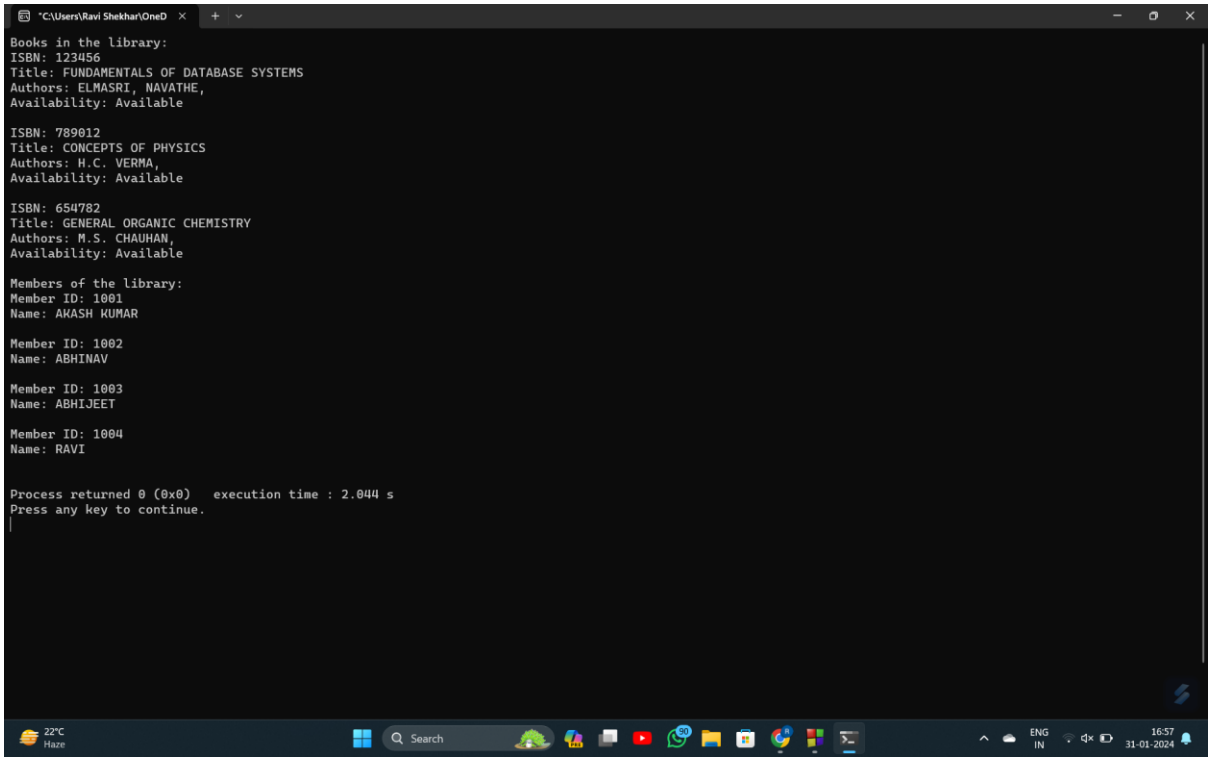
1.

```
#include <stdio.h>
#include <string.h>

// Define structures for Book and Member
struct Author {
    int author_id;
    char name[100];
};

struct Book {
    int isbn;
    char title[100];
    int availability; // 1 for available, 0 for checked out
    struct Author authors[5]; // Assuming a book can have at most 5 authors
    int num_authors;
};
```

```
struct Member {
    int member_id;
    char name[100];
};

// Global arrays to store books and members (limited for simplicity)
struct Book books[100];
struct Member members[100];
int num_books = 0;
int num_members = 0;

// Function to add a new book to the system
void addBook(int isbn, char title[], struct Author authors[], int num_authors) {
    struct Book newBook;
    newBook.isbn = isbn;
    strcpy(newBook.title, title);
    newBook.availability = 1;
    newBook.num_authors = num_authors;
    for (int i = 0; i < num_authors; i++) {
        newBook.authors[i] = authors[i];
    }
    books[num_books++] = newBook;
}

// Function to add a new member to the system
void addMember(int member_id, char name[]) {
```

```
    struct Member newMember;
    newMember.member_id = member_id;
    strcpy(newMember.name, name);
    members[num_members++] = newMember;
}

// Function to display all books in the system
void displayBooks() {
    printf("Books in the library:\n");
    for (int i = 0; i < num_books; i++) {
        printf("ISBN: %d\n", books[i].isbn);
        printf("Title: %s\n", books[i].title);
        printf("Authors: ");
        for (int j = 0; j < books[i].num_authors; j++) {
            printf("%s, ", books[i].authors[j].name);
        }
        printf("\nAvailability: %s\n", (books[i].availability == 1) ? "Available" :
"Checked Out");
        printf("\n");
    }
}

// Function to display all members in the system
void displayMembers() {
    printf("Members of the library:\n");
    for (int i = 0; i < num_members; i++) {
        printf("Member ID: %d\n", members[i].member_id);
```

```
        printf("Name: %s\n", members[i].name);
        printf("\n");
    }
}

int main() {
    // Sample usage
    struct Author authors1[2] = {{1, "ELMASRI"}, {2, "NAVATHE"}};
    struct Author authors2[1] = {{3, "H.C. VERMA"}};
    struct Author authors3[1] = {{4, "M.S. CHAUHAN"}};

    addBook(123456, "FUNDAMENTALS OF DATABASE SYSTEMS", authors1, 2);
    addBook(789012, "CONCEPTS OF PHYSICS", authors2, 1);
    addBook(654782, "GENERAL ORGANIC CHEMISTRY", authors3, 1);

    addMember(1001, "AKASH KUMAR");
    addMember(1002, "ABHINAV");
    addMember(1003, "ABHIJEET");
    addMember(1004, "RAVI");

    displayBooks();
    displayMembers();

    return 0;
}
```



2.

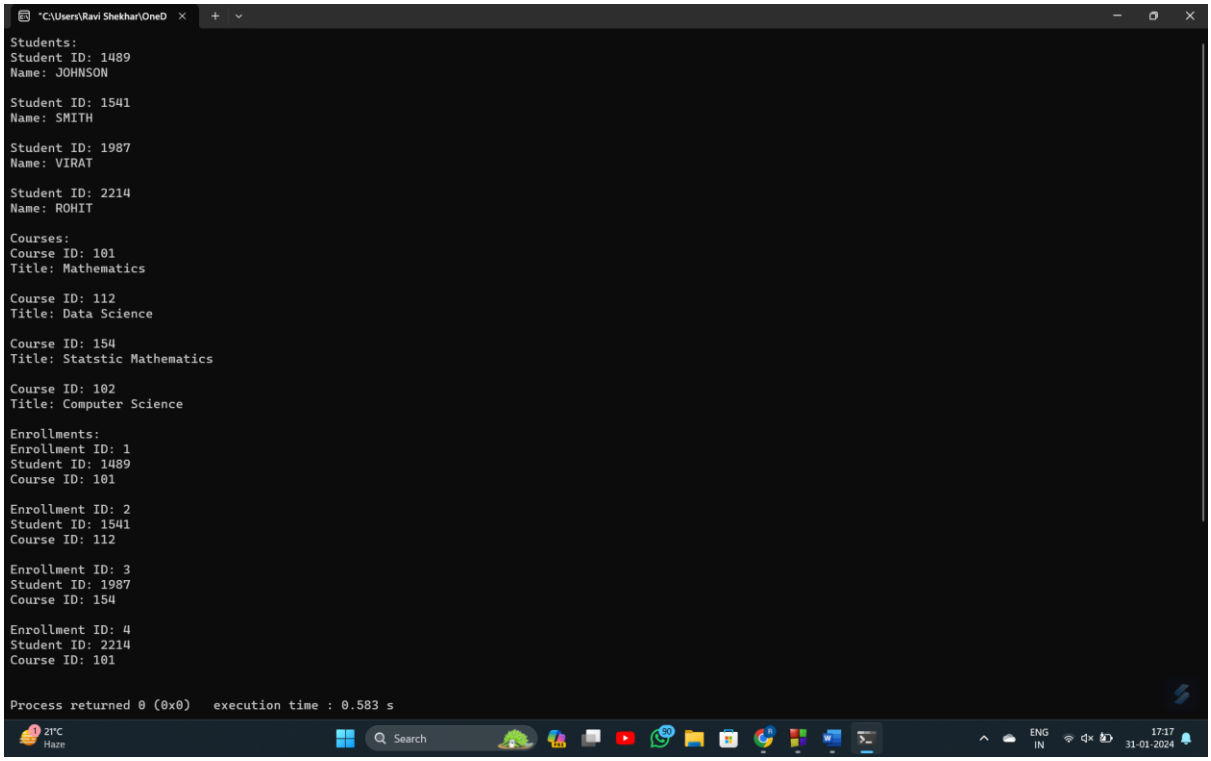
```
#include <stdio.h>
#include <string.h>

// Define structures for Student, Course, and Enrollment
struct Student {
    int student_id;
    char name[100];
};

struct Course {
    int course_id;
    char title[100];
```

```
};

struct Enrollment {
    int enrollment_id;
    int student_id;
    int course_id;
};

// Global arrays to store students, courses, and enrollments (limited for simplicity)
struct Student students[100];
struct Course courses[100];
struct Enrollment enrollments[100];
int num_students = 0;
int num_courses = 0;
int num_enrollments = 0;

// Function to add a new student to the system
void addStudent(int student_id, char name[]) {
    struct Student newStudent;
    newStudent.student_id = student_id;
    strcpy(newStudent.name, name);
    students[num_students++] = newStudent;
}

// Function to add a new course to the system
void addCourse(int course_id, char title[]) {
```

```
    struct Course newCourse;
    newCourse.course_id = course_id;
    strcpy(newCourse.title, title);
    courses[num_courses++] = newCourse;
}

// Function to enroll a student in a course
void enrollStudent(int student_id, int course_id) {
    struct Enrollment newEnrollment;
    newEnrollment.enrollment_id = num_enrollments + 1;
    newEnrollment.student_id = student_id;
    newEnrollment.course_id = course_id;
    enrollments[num_enrollments++] = newEnrollment;
}

// Function to display all students in the system
void displayStudents() {
    printf("Students:\n");
    for (int i = 0; i < num_students; i++) {
        printf("Student ID: %d\n", students[i].student_id);
        printf("Name: %s\n", students[i].name);
        printf("\n");
    }
}

// Function to display all courses in the system
```

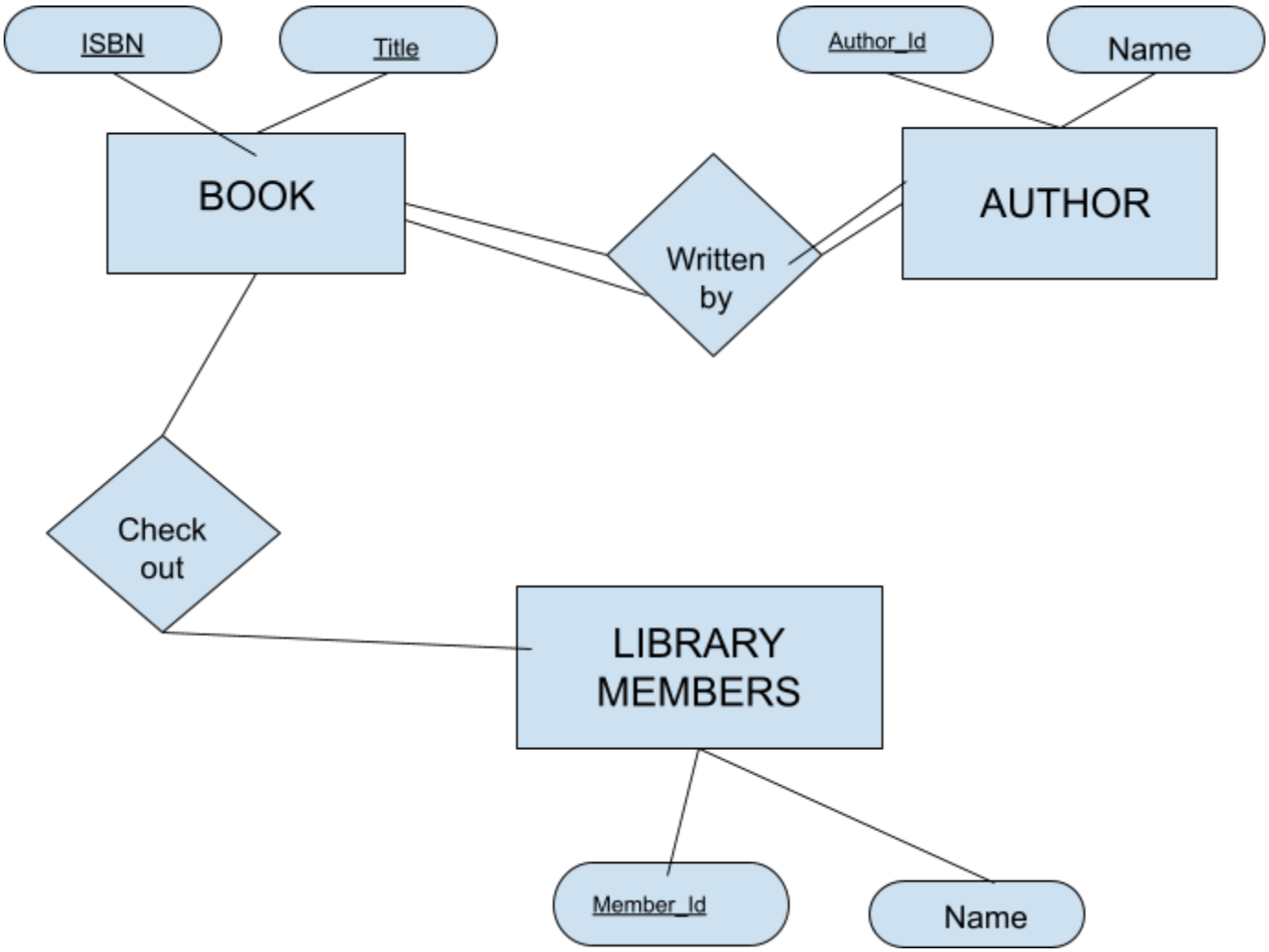
```
void displayCourses() {
    printf("Courses:\n");
    for (int i = 0; i < num_courses; i++) {
        printf("Course ID: %d\n", courses[i].course_id);
        printf("Title: %s\n", courses[i].title);
        printf("\n");
    }
}
// Function to display all enrollments in the system
void displayEnrollments() {
    printf("Enrollments:\n");
    for (int i = 0; i < num_enrollments; i++) {
        printf("Enrollment ID: %d\n", enrollments[i].enrollment_id);
        printf("Student ID: %d\n", enrollments[i].student_id);
        printf("Course ID: %d\n", enrollments[i].course_id);
        printf("\n");
    }
}

int main() {
    // Sample usage
    addStudent(1489, "JOHNSON");
    addStudent(1541, "SMITH");
    addStudent(1987, "VIRAT");
    addStudent(2214, "ROHIT");
```

```
    addCourse(101, "Mathematics");
    addCourse(112, "Data Science");
    addCourse(154, "Statstic Mathematics");
    addCourse(102, "Computer Science");

    enrollStudent(1489, 101);
    enrollStudent(1541, 112);
    enrollStudent(1987, 154);
    enrollStudent(2214, 101);

    displayStudents();
    displayCourses();
    displayEnrollments();
    return 0;
}
```



1.

ER DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM



2.

ER DIAGRAM OF UNIVERSITY DATASYSTEM