Session2

25/07/2024

1)C package to process FILE containing student data.

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

#define MAX_STUDENTS 100
#define MAX_COURSES 4
int chart[7]={10,9,8,7,6,5,0};

typedef struct {
    char course_name[7];
    int credits;
    char grade[2];
} Course;

typedef struct {
    int rollno;
    char name[50];
    char dept[10];
    Course courses[MAX_COURSES];
    int course_count;
    float gpa;
} Student;

Student students[MAX_STUDENTS];
int student_count = 0;

void insertstudent() {
    if (student_count >= MAX_STUDENTS) {
        printf("Student limit reached.\n");
        return;
    }
    Student s;
    printf("Enter roll number: ");
    scanf("%d", &s.rollno);
    printf("Enter name: ");
    scanf("%s", s.name);
    printf("Enter department: ");
    scanf("%s", s.dept);
    printf("Enter number of courses (3 or 4): ");
    scanf("%d", &s.course_count);
    for (int i = 0; i < s.course_count; i++) {
        printf("Enter course%d name: ",i+1);
        scanf("%s", s.courses[i].course_name);
        printf("Enter course%d credits: ",i+1);
        scanf("%d", &s.courses[i].credits);
        printf("Enter course%d grade(S,A,B,C,D,E,F):",i+1);
        scanf("%s", s.courses[i].grade);
    }
    students[student_count++] = s;
    printf("Student record inserted.\n");
}


void calculategpa() {
    for (int i = 0; i < student_count; i++) {
        int totalc = 0;
        int sum = 0;
        for (int j = 0; j < students[i].course_count; j++) {
            totalc += students[i].courses[j].credits;
            char *g=students[i].courses[j].grade;
            int c=(g[0]=='S')?chart[0]:chart[(g[0]-'A')+1];
            sum += students[i].courses[j].credits * c;
        }
        students[i].gpa = (float)sum / totalc;
    }
    printf("GPA calculated for all students.\n");
}

void deletecourse(int rollno) {
    for (int i = 0; i < student_count; i++) {
        if (students[i].rollno == rollno) {
            if (students[i].course_count == 4) {
                students[i].course_count--;
                printf("Last course deregistered for student with roll number %d.
\n", rollno);
            } else {
                printf("Student does not have 4 courses.\n");
            }
            return;
        }
    }
    printf("Student not found.\n");
}
```

```
void insertcourse(int rollno) {
    for (int i = 0; i < student_count; i++) {
        if (students[i].rollno == rollno) {
            if (students[i].course_count == 3) {
                printf("Enter new course name: ");
                scanf("%s",
students[i].courses[students[i].course_count].course_name);
                printf("Enter course credits: ");
                scanf("%d", &students[i].courses[students[i].course_count].credits);
                printf("Enter course grade (S,A,B,C,D,E,F): ");
                scanf("%s", students[i].courses[students[i].course_count].grade);
                students[i].course_count++;
                printf("New course inserted for student with roll number %d.\n",
rollno);
            } else {
                printf("Student has already registered for 4 courses.\n");
            }
            return;
        }
    }
    printf("Student not found.\n");
}

void updatecourse() {
    for (int i = 0; i < 2; i++) {
        int rollno;
        printf("Enter roll number of student to update course: ");
        scanf("%d", &rollno);
        for (int j = 0; j < student_count; j++) {
            if (students[j].rollno == rollno) {
                printf("Registered Courses:\n");
                for (int k = 0; k < students[j].course_count; k++) {
                    printf("%s\t",students[j].courses[k].course_name);
                }
                printf("\n");
                char old_course[20];
                printf("Enter course name to update: ");
                scanf("%s", old_course);
                for (int k = 0; k < students[j].course_count; k++) {
                    if (strcmp(students[j].courses[k].course_name, old_course) == 0) {
                        printf("Enter new course name: ");
                        scanf("%s", students[j].courses[k].course_name);
                        printf("Course updated for student with roll number %d.\n",
rollno);
```

```
                        break;
                    }
                }
            }
        }
    }
}

void upgradegrade(int rollno) {
    for (int i = 0; i < student_count; i++) {
        if (students[i].rollno == rollno) {
            for (int j = 0; j < students[i].course_count; j++) {
                if (students[i].courses[j].grade[0] == 'C') {
                    students[i].courses[j].grade[0] = 'B';
                    printf("Grade upgraded for student with roll number %d.\n",
rollno);
                    return;
                }
            }
        }
    }
    printf("Student or grade not found.\n");
}
void gradereport(int rollno) {
    for (int i = 0; i < student_count; i++) {
        if (students[i].rollno == rollno) {
            printf("Grade report for %s (Roll No: %d):\n", students[i].name,
students[i].rollno);
            for (int j = 0; j < students[i].course_count; j++) {
                printf("Course: %s, Credits: %d, Grade: %s\n",
                       students[i].courses[j].course_name,
                       students[i].courses[j].credits,
                       students[i].courses[j].grade);
            }
            printf("GPA: %.2f\n", students[i].gpa);
            return;
        }
    }
    printf("Student not found.\n");
}

void save() {
    FILE *file = fopen("students.dat", "wb");
    fwrite(&student_count, sizeof(int), 1, file);
    fwrite(students, sizeof(Student), student_count, file);
    printf("Data saved.\n");
    fclose(file);
}
void loadfile() {
    FILE *file = fopen("students.dat", "rb");
    if (file) {
        fread(&student_count, sizeof(int), 1, file);
        fread(students, sizeof(Student), student_count, file);
        fclose(file);
    }
}

int main() {
    loadfile();
    int choice;
    while (1) {
        printf("\nMenu:\n");
        printf("1. Insert student record\n");
        printf("2. Calculate GPA for all students\n");
        printf("3. Delete a course\n");
        printf("4. Insert a new course\n");
        printf("5. Update a course for two students\n");
        printf("6. Upgrade grade\n");
        printf("7. Generate grade report\n");
        printf("8. Save and exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1: insertstudent(); break;
            case 2: calculategpa(); break;
            case 3: {
                int rollno;
                printf("Enter roll number: ");
                scanf("%d", &rollno);
                deletecourse(rollno);
                break;
            }
            case 4: {
                int rollno;
                printf("Enter roll number: ");
                scanf("%d", &rollno);
                insertcourse(rollno);
                break;
            }
            case 5:{
                updatecourse();
                break;
            }

            case 6: {
                int rollno;
                printf("Enter roll number: ");
                scanf("%d", &rollno);
                upgradegrade(rollno);
                break;
            }
            case 7: {
                int rollno;
                printf("Enter roll number: ");
                scanf("%d", &rollno);
                gradereport(rollno);
                break;
            }
            case 8:{
                save();
                return 0;
            }
            default: printf("Invalid choice. Try again please.\n");
        }
    }
    return 0;
}
```

2)SQL DDL Commands

```
Create table student (
    Std_rollno INT PRIMARY KEY,
    Std_name VARCHAR(50),
    Dept CHAR(10),
    C1 CHAR(20),
    C2 CHAR(20),
    C3 CHAR(20),
    C4 CHAR(20)
);


a)insert into student values
      (1, 'Adam', 'CSE', 'DBMS', 'OS', 'Networks', 'AI'),
      (2, 'Bella', 'CSE', 'ARVR', 'OS', 'DBMS', NULL),
      (3, 'Carl', 'CSE', 'FLAT', 'DBMS', 'CC', 'DAA'),
      (4, 'David', 'CSE', 'DSA', 'FLAT', 'EH', 'OS'),
      (5, 'Evans', 'CSE', 'DBMS', 'AI', 'ML', 'OS');

b)alter table student
  DROP COLUMN Course2,
  DROP COLUMN Course3;

c)alter table student
  ADD DoB DATE NOT NULL,
  ADD email VARCHAR2(50) CHECK (email LIKE '%@nitt.edu');

d)alter table student
  MODIFY Course1 VARCHAR2(20);

E)alter table student
  RENAME COLUMN Std_rollno to Std_rno;

F)update student
  SET Course1 = 'OS'
  WHERE Course1 = ‘DBMS';

G)delete FROM student
  WHERE Std_name LIKE 'S%';

H)select * FROM student
  WHERE DoB > TO_DATE('31-12-2005', 'DD-MM-YYYY');

I)drop table student;

J)truncate table student;
```