Name: Reethikka.S.S

Roll_no-106122100

CSE-B

File Processing

1. Develop an implementation package using ‘C’ program to process a FILE containing student

details for the given queries.

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

#define MAX_COURSES 4
#define MAX_STUDENTS 100

typedef struct {
   char course_name[20];
   int credits;
   int grade;
} Course;

typedef struct {
   char roll_no[10];
   char name[50];
   char department[20];
```

```
    Course courses[MAX_COURSES];
    int course_count;
    float gpa;
} Student;

Student students[MAX_STUDENTS];
int student_count = 0;

void read_file() {
    FILE *file = fopen("students.txt", "r");
    if (file == NULL) {
        printf("Could not open file for reading.\n");
        return;
    }

    student_count = 0;
    while (fscanf(file, "%[^,],%[^,],%[^,],", students[student_count].roll_no,
           students[student_count].name, students[student_count].department) != EOF) {
        int i;
        for (i = 0; i < MAX_COURSES; i++) {
            if (fscanf(file, "%[^,],%d,%d,", students[student_count].courses[i].course_name,
                   &students[student_count].courses[i].credits,
&students[student_count].courses[i].grade) == EOF) {
                break;
            }
        }
        students[student_count].course_count = i;
        student_count++;
    }

    fclose(file);
}

void write_file() {
    FILE *file = fopen("students.txt", "w");
    if (file == NULL) {
        printf("Could not open file for writing.\n");
        return;
    }

    for (int i = 0; i < student_count; i++) {
        fprintf(file, "%s,%s,%s,", students[i].roll_no, students[i].name,
students[i].department);
        for (int j = 0; j < students[i].course_count; j++) {
            fprintf(file, "%s,%d,%d,", students[i].courses[j].course_name,
                    students[i].courses[j].credits, students[i].courses[j].grade);
        }
        fprintf(file, "\n");
    }

    fclose(file);
}
```

```
void insert_student() {
    if (student_count >= MAX_STUDENTS) {
        printf("Student limit reached.\n");
        return;
    }

    printf("Enter roll number: ");
    scanf("%s", students[student_count].roll_no);
    printf("Enter name: ");
    scanf("%s", students[student_count].name);
    printf("Enter department: ");
    scanf("%s", students[student_count].department);

    for (int i = 0; i < MAX_COURSES; i++) {
        printf("Enter course %d name: ", i + 1);
        scanf("%s", students[student_count].courses[i].course_name);
        printf("Enter course %d credits: ", i + 1);
        scanf("%d", &students[student_count].courses[i].credits);
        printf("Enter course %d grade: ", i + 1);
        scanf("%d", &students[student_count].courses[i].grade);
        students[student_count].course_count++;
        char more;
        if (i < MAX_COURSES - 1) {
            printf("Do you want to enter more courses? (y/n): ");
            scanf(" %c", &more);
            if (more == 'n') break;
        }
    }

    student_count++;
    write_file();
    printf("Student record inserted.\n");
}

void create_gpa_column() {
    for (int i = 0; i < student_count; i++) {
        float total_points = 0;
        int total_credits = 0;
        for (int j = 0; j < students[i].course_count; j++) {
            total_points += students[i].courses[j].credits * students[i].courses[j].grade;
            total_credits += students[i].courses[j].credits;
        }
        students[i].gpa = total_points / total_credits;
    }
    write_file();
    printf("GPA column created.\n");
}

void deregister_course() {
    char roll_no[10];
    printf("Enter roll number: ");
```

```
scanf("%s", roll_no);

for (int i = 0; i < student_count; i++) {
    if (strcmp(students[i].roll_no, roll_no) == 0) {
        if (students[i].course_count == 4) {
            printf("Enter course name to deregister: ");
            char course_name[20];
            scanf("%s", course_name);

            for (int j = 0; j < students[i].course_count; j++) {
                if (strcmp(students[i].courses[j].course_name, course_name) == 0) {
                    for (int k = j; k < students[i].course_count - 1; k++) {
                        students[i].courses[k] = students[i].courses[k + 1];
                    }
                    students[i].course_count--;
                    write_file();
                    printf("Course deregistered.\n");
                    return;
                }
            }
            printf("Course not found.\n");
            return;
        } else {
            printf("Student does not have 4 courses.\n");
            return;
        }
        }
    }
    printf("Student not found.\n");
}

void insert_course() {
    char roll_no[10];
    printf("Enter roll number: ");
    scanf("%s", roll_no);

    for (int i = 0; i < student_count; i++) {
        if (strcmp(students[i].roll_no, roll_no) == 0) {
            if (students[i].course_count < 4) {
                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: ");
                scanf("%d", &students[i].courses[students[i].course_count].grade);
                students[i].course_count++;
                write_file();
                printf("Course added.\n");
                return;
            } else {
                printf("Student already has 4 courses.\n");
                return;
            }
        }
    }
    printf("Student not found.\n");
}

void update_course_name() {
    for (int i = 0; i < 2; i++) {
        char roll_no[10];
        printf("Enter roll number for student %d: ", i + 1);
        scanf("%s", roll_no);

        for (int j = 0; j < student_count; j++) {
            if (strcmp(students[j].roll_no, roll_no) == 0) {
                printf("Enter old course name: ");
                char old_course_name[20];
                scanf("%s", old_course_name);

                for (int k = 0; k < students[j].course_count; k++) {
                    if (strcmp(students[j].courses[k].course_name, old_course_name) == 0) {
                        printf("Enter new course name: ");
                        scanf("%s", students[j].courses[k].course_name);
                        write_file();
                        printf("Course name updated.\n");
                        return;
                    }
                }
                printf("Course not found.\n");
                return;
            }
        }
        printf("Student not found.\n");
    }
}

void upgrade_grade() {
    char roll_no[10];
    printf("Enter roll number: ");
    scanf("%s", roll_no);

    for (int i = 0; i < student_count; i++) {
        if (strcmp(students[i].roll_no, roll_no) == 0) {
            for (int j = 0; j < students[i].course_count; j++) {
                if (students[i].courses[j].grade == 7) {
                    students[i].courses[j].grade = 8;
                    write_file();
                    printf("Grade upgraded.\n");
                    return;
                }
            }
            printf("No course with grade 7 found.\n");
            return;
        }
    }
    printf("Student not found.\n");
}

void calculate_updated_gpa() {
    char roll_no[10];
    printf("Enter roll number: ");
    scanf("%s", roll_no);

    for (int i = 0; i < student_count; i++) {
        if (strcmp(students[i].roll_no, roll_no) == 0) {
            float total_points = 0;
            int total_credits = 0;
            for (int j = 0; j < students[i].course_count; j++) {
                total_points += students[i].courses[j].credits * students[i].courses[j].grade;
                total_credits += students[i].courses[j].credits;
            }
            students[i].gpa = total_points / total_credits;
            printf("Updated GPA: %.2f\n", students[i].gpa);
            return;
        }
    }
    printf("Student not found.\n");
}
```

```
void generate_grade_report() {
  char identifier[50];
  printf("Enter roll number or name: ");
  scanf("%s", identifier);

  for (int i = 0; i < student_count; i++) {
    if (strcmp(students[i].roll_no, identifier) == 0 || strcmp(students[i].name, identifier) ==
0) {
      printf("Roll No: %s\n", students[i].roll_no);
      printf("Name: %s\n", students[i].name);
      printf("Department: %s\n", students[i].department);
      for (int j = 0; j < students[i].course_count; j++) {
        printf("Course: %s, Credits: %d, Grade: %d\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");
}

int main() {
  int choice;
  read_file();
```

```
while (1) {
  printf("\nMenu:\n");
  printf("1. Insert student record\n");
  printf("2. Create GPA column\n");
  printf("3. Deregister course\n");
  printf("4. Insert course\n");
  printf("5. Update course name\n");
  printf("6. Upgrade grade\n");
  printf("7. Calculate updated GPA\n");
  printf("8. Generate grade report\n");
  printf("9. Exit\n");
  printf("Enter your choice: ");
  scanf("%d", &choice);

  switch (choice) {
    case 1:
      insert_student();
      break;
    case 2:
      create_gpa_column();
      break;
    case 3:
      deregister_course();
      break;
    case 4:
        insert_course();
        break;
      case 5:
        update_course_name();
        break;
      case 6:
        upgrade_grade();
        break;
      case 7:
        calculate_updated_gpa();
        break;
      case 8:
        generate_grade_report();
        break;
      case 9:
        exit(0);
      default:
        printf("Invalid choice.\n");
    }
  }

  return 0;
}
```

a. Insert at least 5 student records.

```
Menu:
```

```
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 1
Enter roll number: 1
Enter name: Anu
Enter department: CSE
Enter course 1 name: OS
Enter course 1 credits: 2
Enter course 1 grade: 8
Do you want to enter more courses? (y/n): y
Enter course 2 name: DBMS
Enter course 2 credits: 3
Enter course 2 grade: 7
Do you want to enter more courses? (y/n): y
Enter course 3 name: C
Enter course 3 credits: 2
Enter course 3 grade: 7
Do you want to enter more courses? (y/n): n
Student record inserted.
```

```
Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 1
Enter roll number: 2
Enter name: Shan
Enter department: CSE
Enter course 1 name: OS
Enter course 1 credits: 3
Enter course 1 grade: 9
Do you want to enter more courses? (y/n): Y
Enter course 2 name: DBMS
Enter course 2 credits: 4
Enter course 2 grade: 8
Do you want to enter more courses? (y/n): Y
Enter course 3 name: CS
Enter course 3 credits: 2
Enter course 3 grade: 7
```

```
Do you want to enter more courses? (y/n): Y
Enter course 4 name: ECS
Enter course 4 credits: 2
Enter course 4 grade: 8
Student record inserted.

Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 1
Enter roll number: 3
Enter name: Ajith
Enter department: CSE
Enter course 1 name: MATH
Enter course 1 credits: 2
Enter course 1 grade: 7
Do you want to enter more courses? (y/n): Y
Enter course 2 name: PHY
Enter course 2 credits: 3
Enter course 2 grade: 8
Do you want to enter more courses? (y/n): Y
Enter course 3 name: ENG
Enter course 3 credits: 3
Enter course 3 grade: 9
Do you want to enter more courses? (y/n): N
Enter course 4 name: CHEM
Enter course 4 credits: 4
Enter course 4 grade: 7
Student record inserted.

Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 1
Enter roll number: 4
Enter name: Jai
Enter department: CSE
Enter course 1 name: MATH
Enter course 1 credits: 2
Enter course 1 grade: 6
Enter course 1 credits: 2
Enter course 1 grade: 6
Do you want to enter more courses? (y/n): Y
Enter course 2 name: PHY
Enter course 2 credits: 3
Enter course 2 credits: 3
Enter course 2 grade: 8
Enter course 2 grade: 8
Do you want to enter more courses? (y/n): Y
Enter course 3 name: ENG
Enter course 3 credits: 2
Enter course 3 grade: 7
Do you want to enter more courses? (y/n): n
Student record inserted.

Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
```

```
9. Exit
Enter your choice: 1
Enter roll number: 5
Enter name: Kirti
Enter department: CSE
Enter course 1 name: OS
Enter course 1 credits: 2
Enter course 1 grade: 7
Do you want to enter more courses? (y/n): Y
Enter course 2 name: DBMS
Enter course 2 credits: 3
Enter course 2 grade: 8
Do you want to enter more courses? (y/n): Y
Enter course 3 name: CN
Enter course 3 credits: 2
Enter course 3 grade: 10
Do you want to enter more courses? (y/n): n
Student record inserted.

Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
```

```
7. Calculate updated GPA

8. Generate grade report

9. Exit
```

b. Create a column ‘GPA’ for all the students

f. Calculate GPA of all students using the GPA formula. Refer the following:

```
Menu:

1. Insert student record

2. Create GPA column

3. Deregister course

4. Insert course

5. Update course name

6. Upgrade grade

7. Calculate updated GPA

8. Generate grade report

9. Exit

Enter your choice: 2

GPA column created.
```

c. For a student with four courses, delete(deregister) a course name.

```
Menu:

1. Insert student record

2. Create GPA column

3. Deregister course
```

```
4. Insert course

5. Update course name

6. Upgrade grade

7. Calculate updated GPA

8. Generate grade report

9. Exit

Enter your choice: 3

Enter roll number: 3

Enter course name to deregister: MATH

Course deregistered.
```

d. For the same student you deleted in ‘c’, insert a new course name.

```
Menu:

1. Insert student record

2. Create GPA column

3. Deregister course

4. Insert course

5. Update course name

6. Upgrade grade

7. Calculate updated GPA

8. Generate grade report

9. Exit

Enter your choice: 4

Enter roll number: 3
Enter new course name: OS

Enter course credits: 2

Enter course grade: 10

Course added.
```

e. Update the name of a course for two different students.

```
Menu:

1. Insert student record

2. Create GPA column

3. Deregister course

4. Insert course

5. Update course name

6. Upgrade grade

7. Calculate updated GPA

8. Generate grade report

9. Exit

Enter your choice: 5

Enter roll number for student 1: 5

Enter old course name: CN

Enter new course name: NETWORKS

Course name updated.
```

g. Upgrade the grade point of a student who has secured ‘7’ in a course

```
Menu:
```

```
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 6
Enter roll number: 2
Grade upgraded.

Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 6
Enter roll number: 5
Grade upgraded.
```

h. Calculate the updated GPA of the student in ‘g’.

```
Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 7
Enter roll number: 5
Updated GPA: 8.57
```

i. Generate a Grade report of a student given the roll no. or name.

```
Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
```

```
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 8
Enter roll number or name: 2
Roll No: 2
Name: Shan
Department: CSE
Course: OS, Credits: 3, Grade: 9
Course: DBMS, Credits: 4, Grade: 8
Course: CS, Credits: 2, Grade: 8
Course: ECS, Credits: 2, Grade: 8
GPA: 8.09

Menu:
1. Insert student record
2. Create GPA column
3. Deregister course
4. Insert course
5. Update course name
6. Upgrade grade
7. Calculate updated GPA
8. Generate grade report
9. Exit
Enter your choice: 9
```

Table:

```
1,Anu,CSE,OS,2,8,DBMS,3,7,C,2,7,

2,Shan,CSE,OS,3,9,DBMS,4,8,CS,2,8,ECS,2,8,

3,Ajith,CSE,PHY,3,8,ENG,3,9,CHEM,4,7,OS,2,10,

4,Jai,CSE,MATH,2,6,PHY,3,8,ENG,2,7,

5,Kirti,CSE,OS,2,8,DBMS,3,8,NETWORKS,2,10,
```

Structured Query Language (SQL) DDL Commands:

1.

```
mysql> CREATE TABLE Student(Std_rollno INT(10),

   -> Std_name VARCHAR(20),

   -> Dept VARCHAR(15),

   -> Course1 CHAR(5),

   -> Course2 CHAR(5),

   -> Course3 CHAR(5),

   -> Course4 CHAR(5),

   -> Email VARCHAR(20),

   -> DOB DATE NOT NULL,

   -> CHECK(Email LIKE '%_@nitt.edu'));

Query OK, 0 rows affected, 1 warning (0.03 sec)


mysql> ALTER TABLE Student ADD PRIMARY KEY(Std_rollno);

Query OK, 0 rows affected (0.15 sec)

Records: 0  Duplicates: 0  Warnings: 0
```

a. Insert at least 5 student records into the Student table.

```
mysql> INSERT INTO Student
VALUES(106123110,'Rahul','CSE','Math','Phy','Chem','CS','106123110@nitt.edu','2005-2-
13');

Query OK, 1 row affected (0.01 sec)


mysql> INSERT INTO Student
VALUES(106123001,'Hiran','ECE','Phy','Math','CS','Chem','106123001@nitt.edu','2003-12-
3');

Query OK, 1 row affected (0.01 sec)


mysql> INSERT INTO Student
VALUES(106123089,'Jaspreet','MECH','Phy','Math','CS','Chem','106123089@nitt.edu','200
3-2-16')

;

Query OK, 1 row affected (0.01 sec)


mysql> INSERT INTO Student
VALUES(106123009,'Rosy','PROD','CS','Phy','Math','Chem','106123009@nitt.edu','2003-7-
6');

Query OK, 1 row affected (0.01 sec)


mysql> INSERT INTO Student
VALUES(106123100,'Rishi','CHEM','EG','Phy','Math','Chem','106123100@nitt.edu','2003-5-
4');

Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT
Std_rollno,Std_name,Dept,Course1,Course2,Course3,Course4,Email,DATE_FORMAT(DO
B,'%d-%m-%Y') AS DOB FROM Stu

dent;

+------------+----------+------+---------+---------+---------+---------+-------------------+------------+

| Std_rollno | Std_name | Dept | Course1 | Course2 | Course3 | Course4 | Email             | DOB
|

+------------+----------+------+---------+---------+---------+---------+-------------------+------------+

|  106123001 | Hiran    | ECE  | Phy     | Math    | CS      | Chem    | 106123001@nitt.edu | 03-12-
2003 |

|  106123009 | Rosy     | PROD | CS      | Phy     | Math    | Chem    | 106123009@nitt.edu | 06-07-
2003 |

|  106123089 | Jaspreet | MECH | Phy     | Math    | CS      | Chem    | 106123089@nitt.edu | 16-
02-2003 |

|  106123100 | Rishi    | CHEM | EG      | Phy     | Math    | Chem    | 106123100@nitt.edu | 04-05-
2003 |

|  106123110 | Rahul    | CSE  | Math    | Phy     | Chem    | CS      | 106123110@nitt.edu | 13-02-
2005 |

+------------+----------+------+---------+---------+---------+---------+-------------------+------------+

5 rows in set (0.00 sec)
```

b. Delete Course2 and Course3 attributes from the Student table

```
mysql> ALTER TABLE Student DROP COLUMN Course2, DROP COLUMN Course3;

Query OK, 0 rows affected (0.07 sec)
```

```
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Student;
+------------+----------+------+---------+---------+--------------------+------------+
| Std_rollno | Std_name | Dept | Course1 | Course4 | Email              | DOB        |
+------------+----------+------+---------+---------+--------------------+------------+
|  106123001 | Hiran    | ECE  | Phy     | Chem    | 106123001@nitt.edu | 2003-12-03 |
|  106123009 | Rosy     | PROD | CS      | Chem    | 106123009@nitt.edu | 2003-07-06 |
|  106123089 | Jaspreet | MECH | Phy     | Chem    | 106123089@nitt.edu | 2003-02-16 |
|  106123100 | Rishi    | CHEM | EG      | Chem    | 106123100@nitt.edu | 2003-05-04 |
|  106123110 | Rahul    | CSE  | Math    | CS      | 106123110@nitt.edu | 2005-02-13 |
+------------+----------+------+---------+---------+--------------------+------------+
5 rows in set (0.00 sec)
```

c. Insert two new columns DoB and email into the Student table.

```
mysql> ALTER TABLE Student ADD DOB1 DATE NOT NULL,
    -> ADD Email1 VARCHAR(25) CHECK(Email1 LIKE '%_@nitt.edu');
ERROR 1292 (22007): Incorrect date value: '0000-00-00' for column 'DOB1' at row 1
```

d. Change Course1 datatype to varchar2.

```
mysql> ALTER TABLE Student MODIFY Course1 VARCHAR(2);
ERROR 1406 (22001): Data too long for column 'Course1' at row 1
```

```
mysql> ALTER TABLE Student MODIFY Course1 VARCHAR(7);

Query OK, 5 rows affected (0.10 sec)

Records: 5  Duplicates: 0  Warnings: 0
```

e. Update the column name ‘Std_rollno’ to ‘Std_rno’.

```
mysql> ALTER TABLE Student CHANGE Std_rollno Std_rno INT;

Query OK, 0 rows affected (0.03 sec)

Records: 0  Duplicates: 0  Warnings: 0


mysql> DESC Student;

+----------+-------------+------+-----+---------+-------+

| Field    | Type        | Null | Key | Default | Extra |

+----------+-------------+------+-----+---------+-------+

| Std_rno  | int         | NO   | PRI | NULL    |       |

| Std_name | varchar(20) | YES  |     | NULL    |       |

| Dept     | varchar(15) | YES  |     | NULL    |       |

| Course1  | varchar(7)  | YES  |     | NULL    |       |

| Course4  | char(5)     | YES  |     | NULL    |       |

| Email    | varchar(20) | YES  |     | NULL    |       |

| DOB      | date        | NO   |     | NULL    |       |

+----------+-------------+------+-----+---------+-------+

7 rows in set (0.01 sec)
```

f. Update all student records who pursue a course named “DBMS” to “OS”

```
mysql> INSERT INTO Student
VALUES(106123111,'Jeeva','CSE','DBMS','Chem','106123111@nitt.edu','2003-9-3');

Query OK, 1 row affected (0.01 sec)


mysql> SELECT * FROM Student;

+-----------+----------+------+---------+---------+--------------------+------------+

| Std_rno   | Std_name | Dept | Course1 | Course4 | Email              | DOB        |

+-----------+----------+------+---------+---------+--------------------+------------+

| 106123001 | Hiran    | ECE  | Phy     | Chem    | 106123001@nitt.edu | 2003-12-03 |

| 106123009 | Rosy     | PROD | CS      | Chem    | 106123009@nitt.edu | 2003-07-06 |

| 106123089 | Jaspreet | MECH | Phy     | Chem    | 106123089@nitt.edu | 2003-02-16 |

| 106123100 | Rishi    | CHEM | EG      | Chem    | 106123100@nitt.edu | 2003-05-04 |

| 106123110 | Rahul    | CSE  | Math    | CS      | 106123110@nitt.edu | 2005-02-13 |

| 106123111 | Jeeva    | CSE  | DBMS    | Chem    | 106123111@nitt.edu | 2003-09-03 |

+-----------+----------+------+---------+---------+--------------------+------------+

6 rows in set (0.00 sec)

mysql> UPDATE Student SET COURSE1='OS' WHERE COURSE1='DBMS';

Query OK, 1 row affected (0.01 sec)

Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Student SET COURSE4='OS' WHERE COURSE4='DBMS';

Query OK, 0 rows affected (0.00 sec)

Rows matched: 0  Changed: 0  Warnings: 0
```

```
mysql> SELECT * FROM Student;

+-----------+----------+------+---------+---------+-------------------+------------+

| Std_rno   | Std_name | Dept | Course1 | Course4 | Email             | DOB        |

+-----------+----------+------+---------+---------+-------------------+------------+

| 106123001 | Hiran    | ECE  | Phy     | Chem    | 106123001@nitt.edu | 2003-12-03 |

| 106123009 | Rosy     | PROD | CS      | Chem    | 106123009@nitt.edu | 2003-07-06 |

| 106123089 | Jaspreet | MECH | Phy     | Chem    | 106123089@nitt.edu | 2003-02-16 |

| 106123100 | Rishi    | CHEM | EG      | Chem    | 106123100@nitt.edu | 2003-05-04 |

| 106123110 | Rahul    | CSE  | Math    | CS      | 106123110@nitt.edu | 2005-02-13 |

| 106123111 | Jeeva    | CSE  | OS      | Chem    | 106123111@nitt.edu | 2003-09-03 |

+-----------+----------+------+---------+---------+-------------------+------------+

6 rows in set (0.00 sec)
```

g. Delete a student record with student name starting with letter ‘S’

```
mysql> INSERT INTO Student
VALUES(106123101,'Sindhu','CSE','Math','Chem','106123101@nitt.edu','2003-1-3');


Query OK, 1 row affected (0.01 sec)


mysql> SELECT * FROM Student;

+-----------+----------+------+---------+---------+-------------------+------------+

| Std_rno   | Std_name | Dept | Course1 | Course4 | Email             | DOB        |

+-----------+----------+------+---------+---------+-------------------+------------+

| 106123001 | Hiran    | ECE  | Phy     | Chem    | 106123001@nitt.edu | 2003-12-03 |

| 106123009 | Rosy     | PROD | CS      | Chem    | 106123009@nitt.edu | 2003-07-06 |
| 106123089 | Jaspreet | MECH | Phy     | Chem    | 106123089@nitt.edu | 2003-02-16 |

| 106123100 | Rishi    | CHEM | EG      | Chem    | 106123100@nitt.edu | 2003-05-04 |

| 106123101 | Sindhu   | CSE  | Math    | Chem    | 106123101@nitt.edu | 2003-01-03 |

| 106123110 | Rahul    | CSE  | Math    | CS      | 106123110@nitt.edu | 2005-02-13 |

| 106123111 | Jeeva    | CSE  | OS      | Chem    | 106123111@nitt.edu | 2003-09-03 |

+-----------+----------+------+---------+---------+--------------------+------------+

7 rows in set (0.00 sec)

mysql> DELETE FROM Student WHERE Std_name LIKE 'S%';

Query OK, 1 row affected (0.01 sec)


mysql> SELECT * FROM Student;

+-----------+----------+------+---------+---------+--------------------+------------+

| Std_rno   | Std_name | Dept | Course1 | Course4 | Email              | DOB        |

+-----------+----------+------+---------+---------+--------------------+------------+

| 106123001 | Hiran    | ECE  | Phy     | Chem    | 106123001@nitt.edu | 2003-12-03 |

| 106123009 | Rosy     | PROD | CS      | Chem    | 106123009@nitt.edu | 2003-07-06 |

| 106123089 | Jaspreet | MECH | Phy     | Chem    | 106123089@nitt.edu | 2003-02-16 |

| 106123100 | Rishi    | CHEM | EG      | Chem    | 106123100@nitt.edu | 2003-05-04 |

| 106123110 | Rahul    | CSE  | Math    | CS      | 106123110@nitt.edu | 2005-02-13 |

| 106123111 | Jeeva    | CSE  | OS      | Chem    | 106123111@nitt.edu | 2003-09-03 |

+-----------+----------+------+---------+---------+--------------------+------------+

6 rows in set (0.00 sec)
```

h. Display all records in which a student has born after the year 2005.

```
mysql> INSERT INTO Student
VALUES(106123101,'Suraj','CSE','Math','Chem','106123101@nitt.edu','2006-1-3');

Query OK, 1 row affected (0.01 sec)


mysql> SELECT * FROM Student;

+-----------+----------+------+---------+---------+--------------------+------------+

| Std_rno   | Std_name | Dept | Course1 | Course4 | Email              | DOB        |

+-----------+----------+------+---------+---------+--------------------+------------+

| 106123001 | Hiran    | ECE  | Phy     | Chem    | 106123001@nitt.edu | 2003-12-03 |

| 106123009 | Rosy     | PROD | CS      | Chem    | 106123009@nitt.edu | 2003-07-06 |

| 106123089 | Jaspreet | MECH | Phy     | Chem    | 106123089@nitt.edu | 2003-02-16 |

| 106123100 | Rishi    | CHEM | EG      | Chem    | 106123100@nitt.edu | 2003-05-04 |

| 106123101 | Suraj    | CSE  | Math    | Chem    | 106123101@nitt.edu | 2006-01-03 |

| 106123110 | Rahul    | CSE  | Math    | CS      | 106123110@nitt.edu | 2005-02-13 |

| 106123111 | Jeeva    | CSE  | OS      | Chem    | 106123111@nitt.edu | 2003-09-03 |

+-----------+----------+------+---------+---------+--------------------+------------+

7 rows in set (0.00 sec)

mysql> SELECT * FROM Student WHERE YEAR(DOB)>2005;

+-----------+----------+------+---------+---------+--------------------+------------+

| Std_rno   | Std_name | Dept | Course1 | Course4 | Email              | DOB        |

+-----------+----------+------+---------+---------+--------------------+------------+

| 106123101 | Suraj    | CSE  | Math    | Chem    | 106123101@nitt.edu | 2006-01-03 |

+-----------+----------+------+---------+---------+--------------------+------------+

1 row in set (0.00 sec)
```

i. Simulate RENAME, COMMENT, TRUNATE and DROP.

```
mysql> RENAME TABLE Student TO Student_details;

Query OK, 0 rows affected (0.03 sec)


mysql> SELECT * FROM Student_details;

+-----------+----------+------+---------+---------+-------------------+------------+

| Std_rno   | Std_name | Dept | Course1 | Course4 | Email             | DOB        |

+-----------+----------+------+---------+---------+-------------------+------------+

| 106123001 | Hiran    | ECE  | Phy     | Chem    | 106123001@nitt.edu | 2003-12-03 |

| 106123009 | Rosy     | PROD | CS      | Chem    | 106123009@nitt.edu | 2003-07-06 |

| 106123089 | Jaspreet | MECH | Phy     | Chem    | 106123089@nitt.edu | 2003-02-16 |

| 106123100 | Rishi    | CHEM | EG      | Chem    | 106123100@nitt.edu | 2003-05-04 |

| 106123101 | Suraj    | CSE  | Math    | Chem    | 106123101@nitt.edu | 2006-01-03 |

| 106123110 | Rahul    | CSE  | Math    | CS      | 106123110@nitt.edu | 2005-02-13 |

| 106123111 | Jeeva    | CSE  | OS      | Chem    | 106123111@nitt.edu | 2003-09-03 |

+-----------+----------+------+---------+---------+-------------------+------------+

7 rows in set (0.01 sec)


mysql> SELECT 2*6; # This is a comment

+-----+

| 2*6 |

+-----+

|  12 |

+-----+

1 row in set (0.00 sec)
```

```
mysql> SELECT 2*6;-- This is a comment
+-----+
| 2*6 |
+-----+
|  12 |
+-----+
1 row in set (0.00 sec)
mysql> SELECT 2/* This is an inline comment */*6;
+------+
| 2 *6 |
+------+
|   12 |
+------+
1 row in set (0.00 sec)

mysql> SELECT 2*
    -> /*
   /*> This is multiline comment
   /*> */
    -> 6;
+-------+
| 2*

6 |
+-------+
|    12 |
+-------+
1 row in set (0.00 sec)

mysql> TRUNCATE TABLE Student_details;
Query OK, 0 rows affected (0.04 sec)

mysql> SELECT * FROM Student_details;
Empty set (0.00 sec)

mysql> DROP TABLE Student_details;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM Student_details;
ERROR 1146 (42S02): Table 'home_work.student_details' doesn't exist
```