

```
#!/bin/bash
```

```
# Function to create the database
```

```
create_database() {  
    echo "Creating new database..."  
    rm -f std.dat      # Delete old database if it exists  
    touch std.dat      # Create a new, empty database  
    echo "New database created successfully."  
}
```

```
# Function to view the database
```

```
view_database() {  
    echo "Viewing database..."  
    if [ -e std.dat ]; then  
        echo -e "Roll No\tName\tAge\tGrade"  
        echo "-----"  
        cat std.dat  
    else  
        echo "Database not yet created."  
    fi  
}
```

```
# Function to insert a record into the database
```

```
insert() {  
    echo "Insert Record"  
    if [ -e std.dat ]; then  
        echo "Enter the Roll No:"
```

```
    read roll
    echo "Enter the Name:"
    read sName
    echo "Enter the Age:"
    read age
    echo "Enter the Grade:"
    read grade
    echo -e "$roll\t$sName\t$age\t$grade" >> std.dat
    echo "Record inserted successfully."
else
    echo "Database does not exist. Please create it first."
fi
}
```

Function to modify a record

```
modify_record() {
    echo "Modify Record"
    if [ -e std.dat ]; then
        echo "Enter the roll number to modify:"
        read roll
        count=$(grep -c "^$roll" std.dat)
        if [ "$count" -eq 0 ]; then
            echo "Roll number $roll not found."
        else
            echo "Enter the new Roll No:"
            read new_roll
            echo "Enter the new Name:"
            read new_name
            echo "Enter the new Age:"
```

```
    read new_age
    echo "Enter the new Grade:"
    read new_grade
    sed -i "s/^$roll\t.*/$new_roll\t$new_name\t$new_age\t$new_grade/" std.dat
    echo "Record modified successfully."
fi
else
    echo "Database does not exist."
fi
}
```

Function to delete a record

```
delete_record() {
    echo "Delete Record"
    if [ -e std.dat ]; then
        echo "Enter the roll number to be deleted:"
        read roll
        count=$(grep -c "^$roll" std.dat)
        if [ "$count" -eq 0 ]; then
            echo "Roll number $roll not found."
        else
            grep -v "^$roll" std.dat > std.temp
            mv std.temp std.dat
            echo "Record deleted successfully."
        fi
    else
        echo "Database does not exist."
    fi
}
```

```
# Function to search for a record
```

```
search_record() {  
    echo "Search Record"  
    if [ -e std.dat ]; then  
        echo "Enter the roll number to search:"  
        read roll  
        result=$(grep "^$roll" std.dat)  
        if [ -z "$result" ]; then  
            echo "Roll number $roll not found."  
        else  
            echo "-----"  
            echo "Student Found:"  
            echo "-----"  
            echo "Roll No : $(echo "$result" | awk '{print $1}')"  
            echo "Name   : $(echo "$result" | awk '{print $2}')"  
            echo "Age    : $(echo "$result" | awk '{print $3}')"  
            echo "Grade  : $(echo "$result" | awk '{print $4}')"  
            echo "-----"  
        fi  
    else  
        echo "Database does not exist."  
    fi  
}
```

```
# Function to display a specific student's result
```

```
display_result() {  
    echo "Display Result"  
    if [ -e std.dat ]; then
```

```

echo "Enter the roll number to view the result:"

read roll

result=$(grep "^$roll" std.dat)

if [ -z "$result" ]; then
    echo "Roll number $roll not found."
else
    echo "-----"
    echo "Student Details:"
    echo "-----"
    echo "Roll No : $(echo "$result" | awk '{print $1}')"
    echo "Name   : $(echo "$result" | awk '{print $2}')"
    echo "Age    : $(echo "$result" | awk '{print $3}')"
    echo "Grade  : $(echo "$result" | awk '{print $4}')"
    echo "-----"
fi
else
    echo "Database does not exist."
fi
}

```

Menu driven interface

```

menu() {
    while true; do
        echo "=====
        echo "    MENU OPTIONS    "
        echo "=====
        echo "1. Create Database"
        echo "2. View Database"
        echo "3. Insert Record"
    done
}

```

```
    echo "4. Modify Record"
    echo "5. Delete Record"
    echo "6. Search Record"
    echo "7. Display Student's Result"
    echo "8. Exit"
    echo "===== "
    echo -n "Please select an option: "
    read choice
    case $choice in
        1) create_database ;;
        2) view_database ;;
        3) insert ;;
        4) modify_record ;;
        5) delete_record ;;
        6) search_record ;;
        7) display_result ;;
        8) echo "Exiting..."; exit 0 ;;
        *) echo "Invalid choice, please try again." ;;
    esac
done
}

# Start the menu
menu
```