**INDEX SHEET**

[**Week 1 [Reading Employee data from file with comma separated fields in each record of employee]**](#_kjirdjkwj59z) **1**

[1. To Write a C program that reads employee records from the file and shows the fields as selected by the user as output. Analyze the problems of maintaining such data in a file.](#_vewuejysj9zn) 1

[**Week 2 [ER Modeling]**](#_esy36cg2j0yw) **2**

[**Experiment – 2: Design a structured database with 4-5 entity sets and possibly relationship sets depending on the relations between entities and prescribe the constraints that need to be enforced for ensuring integrity of data.**](#_w20d3zuodjiu) **2**

[2. To Draw an ER Diagram](#_gcsf07oz4a1n) 2

[**Week 3 [DDL, DML, DQL, DCL]**](#_qpaqr9fuu1ln) **3**

[3. To Demonstrate Data Definition Language (DDL) Commands.](#_my1dqrc1ccw0) 3

[4. To Demonstrate Data Manipulation Language (DML) Commands.](#_bo4mmqh8c219) 6

| Exp# | 1 | Week# | 1 | Date# |  |
| --- | --- | --- | --- | --- | --- |

# Week 1 [Reading Employee data from file with comma separated fields in each record of employee]

Aim:

| 1. To Write a C program that reads employee records from the file and shows the fields as selected by the user as output. Analyze the problems of maintaining such data in a file. |
| --- |

Program:

| #include<stdio.h>  #include<stdlib.h>  struct student  {  int rollno;  char name[30];  float mark;  }stud;  // FUNCTION TO INSERT RECORDS TO THE FILE  void insert()  {  FILE \*fp;  fp = fopen("Record", "a");  printf("Enter the Roll no :");  scanf("%d", &stud.rollno);  printf("Enter the Name :");  scanf("%s",stud.name);  printf("Enter the mark :");  scanf("%f", &stud.mark);  fwrite(&stud, sizeof(stud), 1, fp);  fclose(fp);  }  // FUNCTION TO DISPLAY RECORDS  void disp()  {  FILE \*fp1;  fp1 = fopen("Record", "r");  printf("\nRoll Number\tName\tMark\n\n");  while (fread(&stud, sizeof(stud), 1, fp1))  printf(" %d\t\t%s\t%.2f\n", stud.rollno, stud.name, stud.mark);  fclose(fp1);  }  int avlrollno(int rno)  {  FILE \*fp;  int c = 0;  fp = fopen("Record", "r");  while (!feof(fp))  {  fread(&stud, sizeof(stud), 1, fp);  if (rno == stud.rollno)  {  fclose(fp);  return 1;  }  }  fclose(fp);  return 0;  }  // FUNCTION TO SEARCH THE GIVEN RECORD  void search()  {  FILE \*fp2;  int r, s, avl;  printf("\nEnter the Roll no you want to search :");  scanf("%d", &r);  avl = avlrollno(r);  if (avl == 0)  printf("Roll No %d is not available in the file\n",r);  else  {  fp2 = fopen("Record", "r");  while (fread(&stud, sizeof(stud), 1, fp2))  {  s = stud.rollno;  if (s == r)  {  printf("\nRoll no = %d", stud.rollno);  printf("\nName = %s", stud.name);  printf("\nMark = %.2f\n", stud.mark);  }  }  fclose(fp2);  }  }  // FUNCTION TO DELETE A RECORD  void deletefile()  {  FILE \*fpo;  FILE \*fpt;  int r, s;  printf("Enter the Roll no you want to delete :");  scanf("%d", &r);  if (avlrollno(r) == 0)  printf("Roll no %d is not available in the file\n", r);  else  {  fpo = fopen("Record", "r");  fpt = fopen("TempFile", "w");  while (fread(&stud, sizeof(stud), 1, fpo))  {  s = stud.rollno;  if (s != r)  fwrite(&stud, sizeof(stud), 1, fpt);  }  fclose(fpo);  fclose(fpt);  fpo = fopen("Record", "w");  fpt = fopen("TempFile", "r");  while (fread(&stud, sizeof(stud), 1, fpt))  fwrite(&stud, sizeof(stud), 1, fpo);  printf("\nRECORD DELETED\n");  fclose(fpo);  fclose(fpt);  }  }  // FUNCTION TO UPDATE THE RECORD  void update()  {  int avl;  FILE \*fpt;  FILE \*fpo;  int s, r, ch;  printf("Enter roll number to update:");  scanf("%d", &r);  avl = avlrollno(r);  if (avl == 0)  {  printf("Roll number %d is not Available in the file", r);  }  else  {  fpo = fopen("Record", "r");  fpt = fopen("TempFile", "w");  while (fread(&stud, sizeof(stud), 1, fpo))  {  s = stud.rollno;  if (s != r)  fwrite(&stud, sizeof(stud), 1, fpt);  else  {  printf("\n\t1. Update Name of Roll Number %d", r);  printf("\n\t2. Update Mark of Roll Number %d", r);  printf("\n\t3. Update both Name and Mark of Roll Number %d", r);  printf("\nEnter your choice:");  scanf("%d", &ch);  switch (ch)  {  case 1:  printf("Enter Name:");  scanf("%s",stud.name);  break;  case 2:  printf("Enter Mark : ");  scanf("%f", &stud.mark);  break;  case 3:  printf("Enter Name: ");  scanf("%s",stud.name);  printf("Enter Mark: ");  scanf("%f", &stud.mark);  break;  default:  printf("Invalid Selection");  break;  }  fwrite(&stud, sizeof(stud), 1, fpt);  }  }  fclose(fpo);  fclose(fpt);  fpo = fopen("Record", "w");  fpt = fopen("TempFile", "r");  while (fread(&stud, sizeof(stud), 1, fpt))  {  fwrite(&stud, sizeof(stud), 1, fpo);  }  fclose(fpo);  fclose(fpt);  printf("RECORD UPDATED");  }  }  /\* FUNCTION TO SORT THE RECORD \*/  void sort()  {  int a[20], count = 0, i, j, t, c;  FILE \*fpo;  fpo = fopen("Record", "r");  while (fread(&stud, sizeof(stud), 1, fpo))  {  a[count] = stud.rollno;  count++;  }  c = count;  for (i = 0; i<count - 1; i++)  {  for (j = i + 1; j<count; j++)  {  if (a[i]>a[j])  {  t = a[i];  a[i] = a[j];  a[j] = t;  }  }  }  printf("Roll No.\tName\t\tMark\n\n");  count = c;  for (i = 0; i<count; i++)  {  rewind(fpo);  while (fread(&stud, sizeof(stud), 1, fpo))  {  if (a[i] == stud.rollno)  printf("\n %d\t\t %s \t\t %2f",stud.rollno, stud.name, stud.mark);  }  }  }  // FUNCTION TO CHECK GIVEN ROLL NO IS AVAILABLE //  //FUNCTION TO CHECK THE FILE IS EMPTY OR NOT  int empty()  {  int c = 0;  FILE \*fp;  fp = fopen("Record", "r");  while (fread(&stud, sizeof(stud), 1, fp))  c = 1;  fclose(fp);  return c;  }  // MAIN PROGRAM  void main()  {  int c, emp;  do  {  printf("\n\t---Select your choice---------\n");  printf("\n\t1. INSERT\n\t2. DISPLAY\n\t3. SEARCH");  printf("\n\t4. DELETE\n\t5. UPDATE\n\t6. SORT");  printf("\n\t7. EXIT");  printf("\n\n------------------------------------------\n");  printf("\nEnter your choice:");  scanf("%d", &c);  printf("\n");  switch (c)  {  case 1:  insert();  break;  case 2:  emp = empty();  if (emp == 0)  printf("\nThe file is EMPTY\n");  else  disp();  break;  case 3:  search();  break;  case 4:  deletefile();  break;  case 5:  update();  break;  case 6:  emp = empty();  if (emp == 0)  printf("\n The file is EMPTY\n");  else  sort();  break;  case 7:  exit(1);  break;  default:  printf("\nYour choice is wrong\nPlease try again...\n");  break;  }  } while (c != 7);  } |
| --- |

Code Repository Link:

|  |
| --- |

Output(s):

|  |
| --- |

Observation(s):

|  |
| --- |

| Exp# | 2 | Week# | 2 | Date# |  |
| --- | --- | --- | --- | --- | --- |

# Week 2 [ER Modeling]

# Experiment – 2: Design a structured database with 4-5 entity sets and possibly relationship sets depending on the relations between entities and prescribe the constraints that need to be enforced for ensuring integrity of data.

**Aim:**

| 2. To Draw an ER Diagram |
| --- |

**List of Entities & Attributes:**

|  |
| --- |

**List of Relationships:**

|  |
| --- |

**ER Diagram:**

|  |
| --- |

**EER Diagram:**

|  |
| --- |

**Observation(s):**

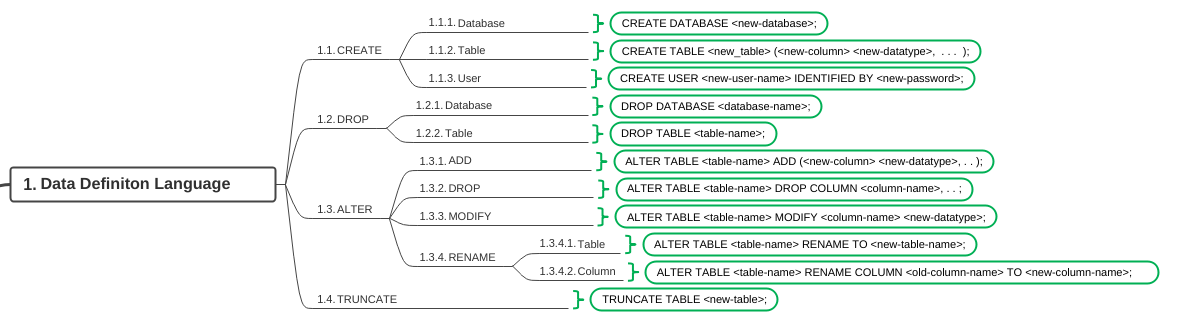
|  |
| --- |

| Exp# | 3 | Week# | 3 | Date# |  |
| --- | --- | --- | --- | --- | --- |

# Week 3 [DDL, DML, DQL, DCL]

Aim :

| 3. To Demonstrate Data Definition Language (DDL) Commands. |
| --- |



SQL Scripts:

| 1.1 CREATE | 1.1.1 Database : Create a database with your *‘roll number’* |
| --- | --- |
| Syntax | CREATE DATABASE <<database-name>>; |
| Usage | It is used to create an empty database in mysql server. |
| SQL Command/ Query(s) | create database dummy;  show databases; |
| Output |  |
| Observation/ Remark(s) | We have used a DQL (show databases) to list the newly created database(20331A0\*\*\*) |

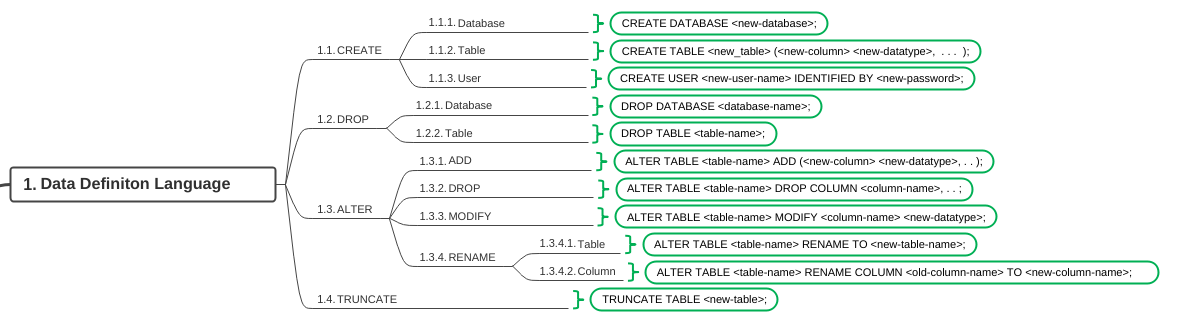
| 1.1 CREATE | 1.1.2 Table : Create a Table *‘Index Sheet’* |
| --- | --- |
| Syntax | CREATE TABLE <new\_table> (<new-column> <new-datatype>, . . . ); |
| Usage | It is used to create an empty table with specified columns in mysql server. |
| SQL Command/ Query(s) | use 20331A5z0;  create table indexSheet(expid int(10) unique, week int(10),experiment varchar(60));  desc indexSheet; |
| Output |  |
| Observation/ Remark(s) | * To create a table, we need to select a database using ‘use’ DQL command. * To display the structure of the table we have used ‘desc’ DQL command. |

| 1.1 CREATE | 1.1.3 User : Create a user with your *‘roll-number’* |
| --- | --- |
| Syntax | CREATE USER <new-user-name> IDENTIFIED BY <new-password>; |
| Usage | It is used to create a user with a password. |
| SQL Command/ Query(s) | use 20331A5z0;  create table indexSheet(expid int(10) unique, week int(10),experiment varchar(60));  desc indexSheet; |
| Output |  |
| Observation/ Remark(s) | * To create a table, we need to select a database using ‘use’ DQL command. * To display the structure of the table we have used ‘desc’ DQL command. |

| Exp# | 4 | Week# | 3 | Date# |  |
| --- | --- | --- | --- | --- | --- |

Aim :

| 4. To Demonstrate Data Manipulation Language (DML) Commands. |
| --- |



SQL Scripts:

| 1.1 CREATE | 1.1.1 Database : Create a database with your *‘roll number’* |
| --- | --- |
| Syntax | CREATE DATABASE <<database-name>>; |
| Usage | It is used to create an empty database in mysql server. |
| SQL Command/ Query(s) | create database dummy;  show databases; |
| Output |  |
| Observation/ Remark(s) | We have used a DQL (show databases) to list the newly created database(20331A0\*\*\*) |

| 1.1 CREATE | 1.1.2 Table : Create a Table *‘Index Sheet’* |
| --- | --- |
| Syntax | CREATE TABLE <new\_table> (<new-column> <new-datatype>, . . . ); |
| Usage | It is used to create an empty table with specified columns in mysql server. |
| SQL Command/ Query(s) | use 20331A5z0;  create table indexSheet(expid int(10) unique, week int(10),experiment varchar(60));  desc indexSheet; |
| Output |  |
| Observation/ Remark(s) | * To create a table, we need to select a database using ‘use’ DQL command. * To display the structure of the table we have used ‘desc’ DQL command. |

| 1.1 CREATE | 1.1.3 User : Create a user with your *‘roll-number’* |
| --- | --- |
| Syntax | CREATE USER <new-user-name> IDENTIFIED BY <new-password>; |
| Usage | It is used to create a user with a password. |
| SQL Command/ Query(s) | use 20331A5z0;  create table indexSheet(expid int(10) unique, week int(10),experiment varchar(60));  desc indexSheet; |
| Output |  |
| Observation/ Remark(s) | * To create a table, we need to select a database using ‘use’ DQL command. * To display the structure of the table we have used ‘desc’ DQL command. |