आईएफटीएम विश्वविद्यालय, मुरादाबाद, उत्तर प्रदेश

IFTM University, Moradabad, Uttar Pradesh

NAAC ACCREDITED

E-Content

IFTM University, Moradabad

IFTM UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND APPLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS

Lab Manual

on

BUSINESS DATA PROCESSING LAB (BCAPL-201)

Faculty Name: Shelly Bhardwaj

Session-2021-2022

Index

Program Number: 01

Objective: Demonstration of MS word processing skills.
Procedure:

Word-Processing Skills:-

Word-Processing Essentials

The most popular word-processing program is Microsoft Word, part of the Microsoft Office Suite. Most college computer labs have this program available, and you can sometimes purchase it for a reduced rate through your college bookstore or campus technology center.

Free Alternatives to Word

Word is the most common word processor, but it's expensive, especially if it didn't come with your computer. Some classes will explicitly require you to use Microsoft Word for your classwork. Otherwise, you're free to use whatever program you wish. Google's Google Docs and Apple's Pages, are two other widely-used examples. Here are some **free** options to explore as an alternative to Word:

- **Google Drive**. Connected to a Gmail / Google personal account, this flexible tool lets you access your saved files from any internet-connected device. You can also download files to work on offline, and they will automatically sync when you go online again.
- **Office Online**. You'll need to register with an account, but can then access your saved files from any internet-connected device.
- **OpenOffice**. This is software you download on your computer, so you don't have to be online to use it.
- **LibreOffice**. Similar to OpenOffice, you download this software directly to your computer.

Using Microsoft Word

When you first open Microsoft Word, you'll be given the option to choose from various templates. Some of these may be useful for your classes, particularly newsletters, brochures, and resumes, but you will most frequently be working in a simple Word Document. Once inside the document, familiarize yourself with the various tools to help you format your paper.

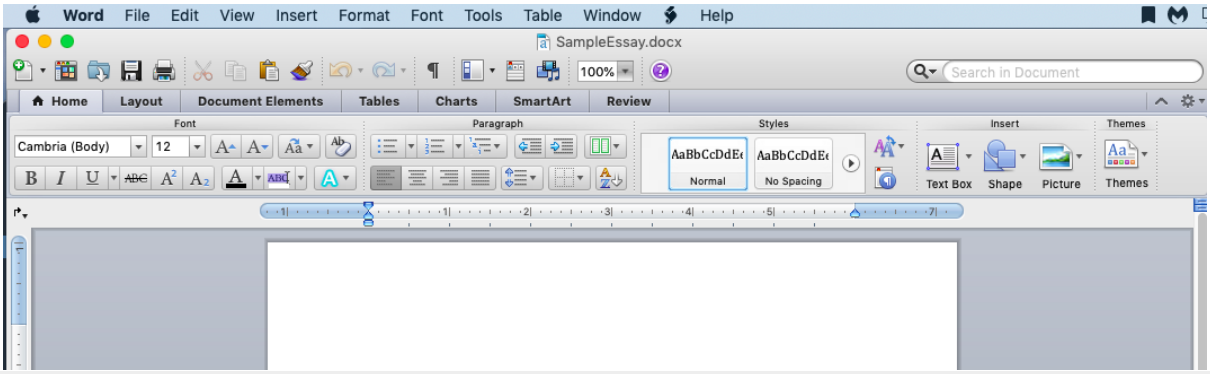


Figure 1. Take some time to investigate the various tools and options inside of Microsoft Word and use these to your advantage when you write.

- Visit the "Format tab" or the "Layout" tab in the toolbar if you need to adjust the sizing of your font or margins
- Use the home tab of the toolbar to adjust the font, font size, and address basic formatting needs, such as adding a numbered list, bullet points, columns, or images.
- Use the tools tab to check for spelling and grammar issues, track changes during editing, or check the word count (although you can also highlight text and see a tally of the word count in the bar at the bottom of the document).

Visit each of the links below to review important features and uses of Microsoft Word:

Basic Skills

- **Creating and Opening Documents**
- **Saving and Sharing:** The ever-important "Save" feature is going to be your new best friend in college.
- **Text Basics:** Cutting, copying, pasting, and deleting are all reviewed here. "Find and replace" is a tool that will be particularly handy for revising documents.
- **Formatting Text**: Making your document look attractive is one of the most fun parts of using a word processor. This video demonstrates quick ways to change your text's appearance.
- **Page Layout:** Your professor may have specific instructions for how she wants you to format the documents you write for her class. If that's the case, review how to change layout and formatting settings here.
- **Printing**

Intermediate Skills

- **Line and Paragraph Spacing**: Having trouble getting things to line up the way you want them to on the page? This video shows how to simplify the appearance of your text with a couple of clicks.

- **Spell Check and Grammar Check**: Probably everyone's favorite tool since the invention of computers is automatic spell check. Grammar check is also quite useful. Though neither tool is perfect, both will provide you a good leg up in the proofreading process.
- **Headers, Footers**, and **Page Numbers**: Your instructor will often ask you to include page numbers on your document, along with some specific formatting procedures. MLA and APA document formats, for instance, both rely upon the use of Headers. See how to insert these easily here.
- **Track Changes**: Some built-in tools allow you to add comments on a draft, which is useful for doing peer review, or making notes to yourself as you build a project. Your instructor might also add notes to your essay document when he grades it, so it's useful to know how to turn on Track Changes so you don't miss his input.
- **Word Count**: Many writing assignments you have in college will ask for a particular word count range (such as a 500–750 word essay assignment). It's useful to know how to easily locate the word count in a document you've created.

Advanced Skills

- **Hanging Indents**: MLA and APA bibliography pages use a special type of indent, called a "hanging indent." Where a normal paragraph indents the first line but not any others, a hanging indent paragraph DOESN'T indent the first line, but DOES indent all the others. Luckily, it's very easy to have your word processor do the hard work of this type of formatting for you, as this video demonstrates.
- **Bibliography Pages**: Speaking of bibliographies (or Works Cited pages, or References pages), did you know that many word processors have ways to help you create those quickly?
- **Alphabetizing Bibliographies (and other lists)**: One more bibliography tip…if you create your citation list as you use sources, you'll need to put these in alphabetical order at the end. An easy way to do that is to use the Sort feature in your word processor. This video demonstrates that in Word 2016.
- **Inserting Pictures**: Many college projects will require you to include visuals in your essays. The following video addresses how to add an image and then how to get the text around it to behave properly afterward.
- **Change Default Settings**: If the first thing you do each time you open a new document, is to change your font size or style, as well as readjust your margins, then you probably will save time by changing the default settings so it starts just the way you like it. (This video is for Word 2013)

Using Google Docs

Working in Google Docs is similar to working in Microsoft Word. The only requirement is that you need a Google login to save and share your documents, and most college and university email addresses

come synced with a Google log-in. You can access Google Docs from inside your **Google Drive**, going directly to the website, or by clicking on the Drive button in the upper-right hand corner of your Google account.

Inside of the Google Drive, you can create Google Docs, Google Sheets (like an Excel document), Google Slides (similar to Microsoft PowerPoint), and more. One advantage of Google Docs is that they are made to be collaborative, so it's very easy to share materials with classmates and your instructor. Note that some LMSs will require you to download your Google Doc as a PDF or Word document before you upload it; the video below explains how to do that (.37).

Once inside of a Google Doc, you'll see that it is very similar to Microsoft Word. If there are features you don't see but would like to include in your document, it may be worth searching for "add-ons" (or searching YouTube for tutorials) to enhance your document.

Program Number: 02

Objective: Write a program to implement merge sort in C.

Procedure:

```
#include <stdio.h>

/* Function to merge the subarrays of a[] */
void merge(int a[], int beg, int mid, int end)
{
   int i, j, k;
   int n1 = mid - beg + 1;
   int n2 = end - mid;

   int LeftArray[n1], RightArray[n2]; //temporary arrays

   /* copy data to temp arrays */
   for (int i = 0; i < n1; i++)
   LeftArray[i] = a[beg + i];
   for (int j = 0; j < n2; j++)
   RightArray[j] = a[mid + 1 + j];

   i = 0; /* initial index of first sub-array */
   j = 0; /* initial index of second sub-array */
   k = beg;  /* initial index of merged sub-array */

   while (i < n1 && j < n2)
   {
      if(LeftArray[i] <= RightArray[j])
      {
         a[k] = LeftArray[i];
         i++;
      }
      else
      {
         a[k] = RightArray[j];
         j++;
      }
      k++;
   }
   while (i<n1)
   {
      a[k] = LeftArray[i];
      i++;
      k++;
   }
```

```
    while (j<n2)
    {
        a[k] = RightArray[j];
        j++;
        k++;
    }
}

void mergeSort(int a[], int beg, int end)
{
    if (beg < end)
    {
        int mid = (beg + end) / 2;
        mergeSort(a, beg, mid);
        mergeSort(a, mid + 1, end);
        merge(a, beg, mid, end);
    }
}

/* Function to print the array */
void printArray(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

int main()
{
    int a[] = { 12, 31, 25, 8, 32, 17, 40, 42 };
    int n = sizeof(a) / sizeof(a[0]);
    printf("Before sorting array elements are - \n");
    printArray(a, n);
    mergeSort(a, 0, n - 1);
    printf("After sorting array elements are - \n");
    printArray(a, n);
    return 0;
}
```

Output:

```
Before sorting array elements are -
12 31 25 8 32 17 40 42
After sorting array elements are -
8 12 17 25 31 32 40 42
```

Program Number: 03

Objective: To create a table using table menu with a given no of rows & columns in excel.

Procedure:

a. First create a table using table menu with a given no rows & columns

b. Then fill the cells of table using data provided.

c. Then apply merge the first row into one cell.

d. Then merge the second row into one cell.

e. Apply border to the table.

Berhampur University B.Ed. Merit List				
Sl. no	Name of students	father's name	Age of the student	phone number
1	Anita Kumari	Suresh Kumar	18	9765473483
2	Rahul Khanna	Devesh Khanna	17	8745200764
3	Pooja rani	Mohan lal	18	9873027450
4	Kavita Sahoo	Rupesh Sahoo	19	8794503298
5	Sangeeta Mallik	Pritam Mallik	18	7600348750
6	Dipti swain	Rajesh swain	17	8704657385
7	Torun Dash	Sambhab Dash	19	8765490086
8	Meghna Agrawal	Kamesh Agrawal	17	9845678910

Program Number: 04

Objective: Create a table "Student Result" with some conditions in excel.

Procedure:

a. First insert a table in the Excel Sheet by the insert tab so that you can write the student's information.

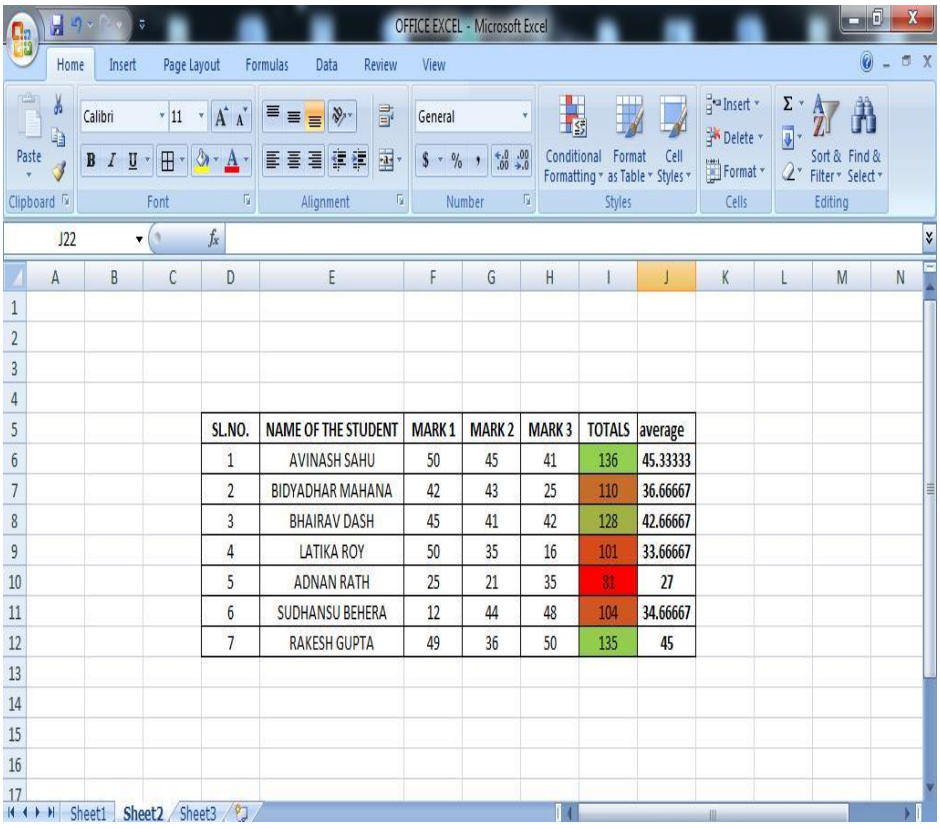
b. Now, give the headings – SL NO, Name, Mark 1, Mark 2, Mark 3, Total etc.

c. Fill up the data in the columns you have to enter.

d. Calculate the sum of the marks of each student via 'Auto sum' option. e. Highlight the highest total mark in green color and lowest one in red color.

f. Find the average of the marks of each student in an individual cell

SAMPLE OUTPUT:



SL.NO.	NAME OF THE STUDENT	MARK 1	MARK 2	MARK 3	TOTALS	average
1	AVINASH SAHU	50	45	41	136	45.33333
2	BIDYADHAR MAHANA	42	43	25	110	36.66667
3	BHAIRAV DASH	45	41	42	128	42.66667
4	LATIKA ROY	50	35	16	101	33.66667
5	ADNAN RATH	25	21	35	81	27
6	SUDHANSU BEHERA	12	44	48	104	34.66667
7	RAKESH GUPTA	49	36	50	135	45

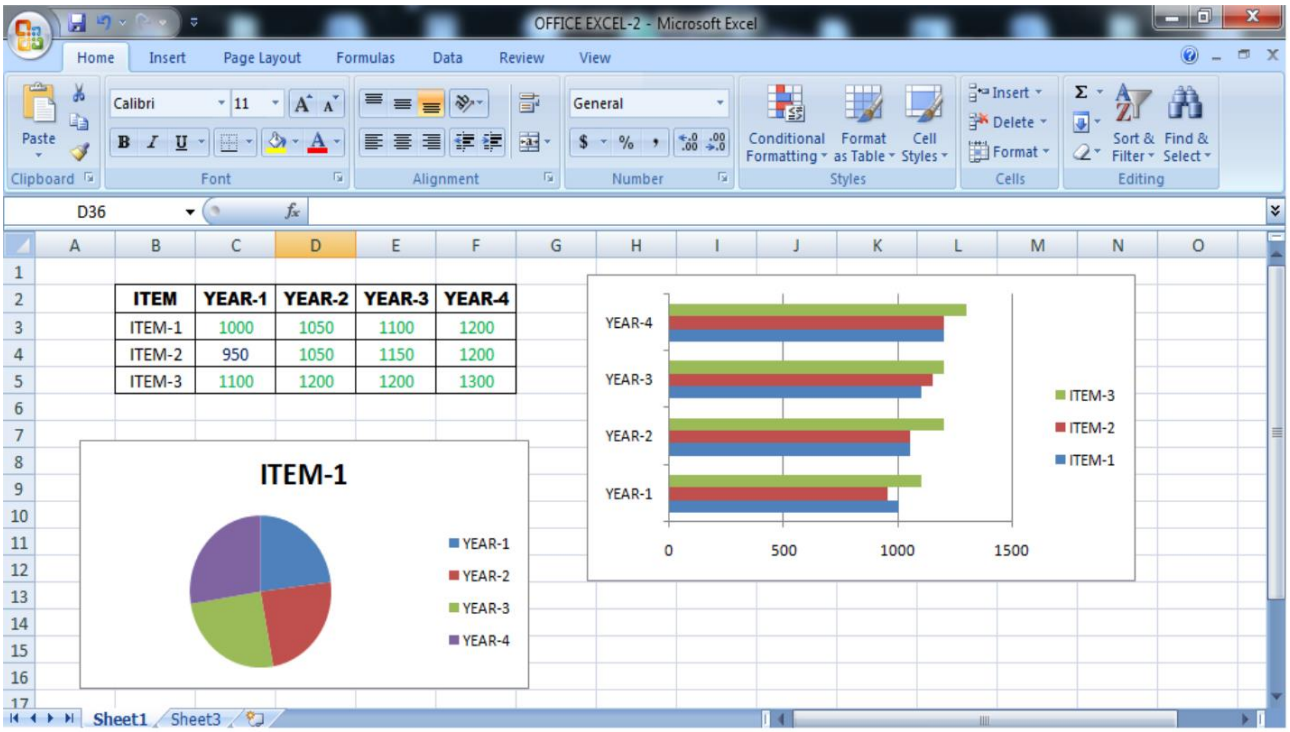
Program Number: 05

Objective: Create a table “Sales” with some data provided in excel.

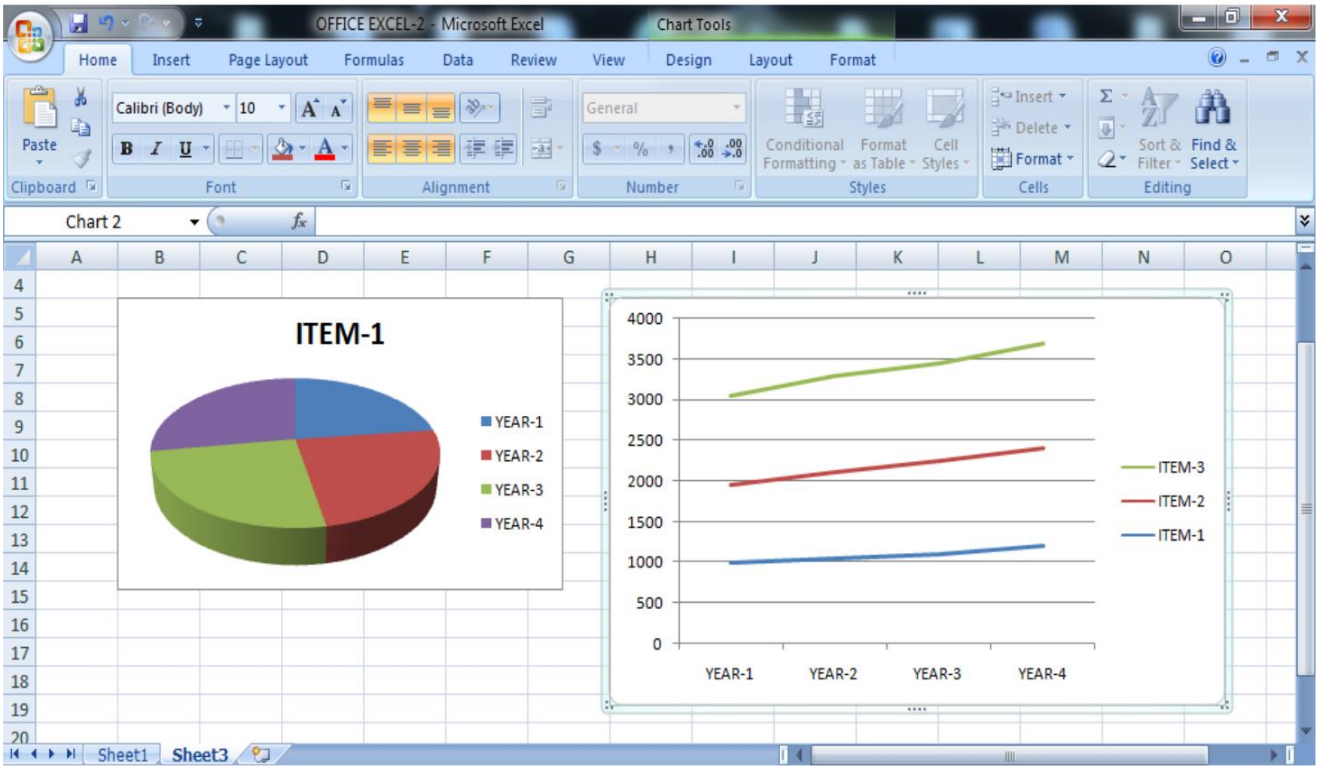
Procedure:

a. Create a table using five columns & four rows.
b. Enter the data given as provided
c. Go to insert tab, in charts groups, you just have to select the specified charts as given in no. a, b, and c.
d. Set a new rule for highlighting the sales which are greater than or equal to 1000
e. To do so, go to conditional tab. There, click more rules option. There you can change all the necessary options.

SAMPLE OUTPUT:



ITEM	YEAR-1	YEAR-2	YEAR-3	YEAR-4
ITEM-1	1000	1050	1100	1200
ITEM-2	950	1050	1150	1200
ITEM-3	1100	1200	1200	1300



Program Number: 06

Objective: What is SQL? Also explain rules and data types of SQL.

Procedure:

SQL(Structure Query Language)

SQL stands for Structured Query Language. Structure Query Language (SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, SQL server, Oracle, Informix, Sybase, MS Access, POstgres) use **SQL** as the standard database query language. It enables a user to create, read, update and delete relational databases and tables.

Rules:

SQL follows the following rules:

- Structure query language is not case sensitive. Generally, keywords of SQL are written in uppercase.
- Statements of SQL are dependent on text lines. We can use a single SQL statement on one or multiple text line.
- Using the SQL statements, we can perform most of the actions in a database.
- SQL depends on tuple relational calculus and relational algebra.

Characteristics of SQL

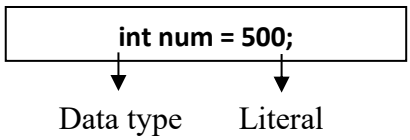
- SQL is easy to learn.
- SQL is used to access data from relational database management systems.
- SQL can execute queries against the database.
- SQL is used to describe the data.
- SQL is used to define the data in the database and manipulate it when needed.
- SQL is used to create and drop the database and table.
- SQL is used to create a view, stored procedure, function in a database.
- SQL allows users to set permissions on tables, procedures, and views.

SQL Data type and literals

SQL Data Type is an attribute that specifies the type of data of any object. Each column, variable and expression has a related data type in SQL. We can use these data types while creating our tables. We can choose a data type for a table column based on our requirement.

- SQL Data type is used to define the values that a column can contain.
- Every column is required to have a name and data type in the database table.

Data types are those which specify the type of data represented by the variable and literal is the value that is stored in to the variable.



Datatype	Use	Literal
INT	used for columns which will store integer values.	050 ,78, -14, 0 , +32767
FLOAT	used for columns which will store float values.	6.2, 2.9, 55.89
VARCHAR	used for columns which will be used to store characters and integers, basically a string.	'Hello world!'
DATE	used for columns which will store date values.	'1978-12-25';
TIME	used for columns which will store time values.	‘12:01:01';
TEXT	used for columns which will store text which is generally long in length. For example, if you create a table for storing profile information of a social networking website, then for **about me** section you can have a column of type TEXT	‘Hello Myself John. I am a database designer. ’

Program Number: 07

Objective: What is DDL? Explain its Commands.

Procedure:

Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

1.	**CREATE** It is used to create a new table in the database. **Syntax:** CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES [,....]); **Example:** CREATE TABLE Student(St_Name VARCHAR2(20), Stu_id int (20), Email VARCHAR2(100), DOB DATE);
2.(a)	**ALTER:** It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute. **Syntax:** To add a new column in the table ALTER TABLE *table_name* ADD *column_name datatype*;

	EXAMPLE ALTER TABLE Students ADD Address varchar(255);
(b)	To delete a column in a table ALTER TABLE *table_name* DROP COLUMN *column_name*; Example ALTER TABLE Students DROP COLUMN address;
(c)	To change the data type of a column in a table ALTER TABLE *table_name* ALTER COLUMN *column_name datatype*; Example ALTER TABLE Persons ALTER COLUMN DateOfBirth year;
3.	**DROP:** It is used to drop an existing table in a database. It is used to delete both the schema/structure and record stored in the table. DROP is permanently lost and it cannot be rolled back. **Syntax** DROP TABLE *table_name*; Example DROP TABLE Employees;
4.	**TRUNCATE:** It is used to delete all the rows from the table and free the space containing the table. **Syntax:**

	TRUNCATE TABLE *table_name*;

Program Number: 08

Objective: What is DML? Explain its Commands.

Procedure:

Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

1.(a)	**INSERT:** The INSERT statement is a SQL query. It is used to insert data into the row of a table. **Syntax:** INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...) VALUES (*value1*, *value2*, *value3*, ...); Example INSERT INTO Customers (CustomerName, City, Country) VALUES ('Ramesh', Moradabad','India');
(b)	For adding values for all the columns of the table, we do not need to specify the column names in the SQL query. INSERT INTO *table_name* VALUES (*value1*, *value2*, *value3*, ...);
2.	**UPDATE:** This command is used to update or modify the value of a column in the table.

	Syntax: UPDATE *table_name* SET *column1* = *value1*, *column2* = *value2*, ... WHERE *condition*; Example UPDATE Students SET Stu_Add = 'Rampur', Phone= '4568233545' WHERE St_rollno = 184524552;
3.	**DELETE:** It is used to remove one or more row from a table. **Syntax:** DELETE FROM *table_name* WHERE *condition*; Example DELETE FROM Students WHERE St_Id='1820530255';

Program Number: 09

Objective: What is DQL? Explain its Commands.

Procedure:

Data Query Language

DQL is used to fetch the data from the database.

It uses only one command:

- SELECT

1.	SELECT: The SELECT statement is used to select data from a database. Syntax: SELECT *column1*, *column2, ...* FROM *table_name*; **Example** SELECT *stu_rollno ,stu_name*, *stu_add* FROM *student*; Or SELECT * FROM *student*; Or SELECT student_name FROM student WHERE age > 20; Select stu_roll, stu_name FROM student WHERE course=BCA;

Program Number: 10

Objective: Write a SQL query to create a table.

Procedure:

```
mysql> CREATE TABLE  Students
    -> (
    -> St_roll_no int,
    -> St_FName varchar(50),
    -> St_LName varchar(50),
    -> Address varchar(50),
    -> City varchar(50),
    -> Email_id varchar(50)
    -> );
```

Query OK, 0 rows affected (0.14 sec)

Output

```
mysql> desc students;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| St_roll_no | int(11)     | YES  |     | NULL    |       |
| St_FName   | varchar(50) | YES  |     | NULL    |       |
| St_LName   | varchar(50) | YES  |     | NULL    |       |
| Address    | varchar(50) | YES  |     | NULL    |       |
| City       | varchar(50) | YES  |     | NULL    |       |
| Email_id   | varchar(50) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Program Number: 11

Objective: Write a SQL query to insert a new row in table.

Procedure:

mysql> insert into students

-> values

-> (1750255, 'Rohan', 'Sharma', 'H.no-135, Near Pillikothi', 'Moradabad', 'rohansharma01@gmail.com');

Query OK, 1 row affected (0.00 sec)

Output

```
mysql> select * from students;
+------------+----------+----------+--------------------------+-----------+------------------------+
| St_roll_no | St_FName | St_LName | Address                  | City      | Email_id               |
+------------+----------+----------+--------------------------+-----------+------------------------+
|    1750255 | Rohan    | Sharma   | H.no-135, Near Pillikothi | Moradabad | rohansharma01@gmail.com |
+------------+----------+----------+--------------------------+-----------+------------------------+
1 row in set (0.00 sec)
```

Program Number: 12

Objective: Write a SQL query to insert a new value in table.

Procedure:

```
mysql> insert into students
    -> (St_FName,St_LName)
    -> values
    -> ('Mohan','Singh');
Query OK, 1 row affected (0.13 sec)
```

Output:

```
mysql> select * from students;
+------------+----------+----------+--------------------------+-----------+------------------------+
| St_roll_no | St_FName | St_LName | Address                  | City      | Email_id               |
+------------+----------+----------+--------------------------+-----------+------------------------+
|    1750255 | Rohan    | Sharma   | H.no-135, Near Pillikothi | Moradabad | rohansharma01@gmail.com |
|       NULL | Mohan    | Singh    | NULL                     | NULL      | NULL                   |
+------------+----------+----------+--------------------------+-----------+------------------------+
2 rows in set (0.06 sec)
```

Program Number: 13

Objective: Write a SQL query to insert a column in existing table.

Procedure:

mysql> alter table students

-> add column phone_no int;

Query OK, 3 rows affected (0.59 sec)

Output

```
mysql> desc students;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| St_roll_no | int(11)     | YES  |     | NULL    |       |
| St_FName   | varchar(50) | YES  |     | NULL    |       |
| St_LName   | varchar(50) | YES  |     | NULL    |       |
| Address    | varchar(50) | YES  |     | NULL    |       |
| City       | varchar(50) | YES  |     | NULL    |       |
| Email_id   | varchar(50) | YES  |     | NULL    |       |
| phone_no   | int(11)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.06 sec)
```

Program Number: 14

Objective: Write a SQL query to drop a column from existing table.

Procedure:

```
mysql> alter table students
    -> drop column Email_id;
Query OK, 3 rows affected (0.08 sec)
```

Output

```
mysql> desc students;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| St_roll_no | int(11)     | YES  |     | NULL    |       |
| St_FName   | varchar(50) | YES  |     | NULL    |       |
| St_LName   | varchar(50) | YES  |     | NULL    |       |
| Address    | varchar(50) | YES  |     | NULL    |       |
| City       | varchar(50) | YES  |     | NULL    |       |
| phone_no   | int(11)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Program Number: 15

Objective: Write a SQL query to update a value in table.

Procedure:

```
mysql> update students
    -> set St_LName= 'Shrivastava'
    -> where St_roll_no=1750255;
Query OK, 1 row affected (0.03 sec)
```

Output

```
mysql> select * from students;
+------------+----------+-------------+---------------------------+-----------+----------+
| St_roll_no | St_FName | St_LName    | Address                   | City      | phone_no |
+------------+----------+-------------+---------------------------+-----------+----------+
|    1750255 | Rohan    | Shrivastava | H.no-135, Near Pillikothi | Moradabad |        0 |
|       NULL | Mohan    | Singh       | NULL                      | NULL      |     NULL |
+------------+----------+-------------+---------------------------+-----------+----------+
2 rows in set (0.00 sec)
```

Program Number: 16

Objective: Write a SQL query to delete a value from a table.

Procedure:

```
mysql> delete from students
    -> where St_FName='Mohan';
```

Query OK, 1 row affected (0.00 sec)

Output

```
mysql> select * from students;
+------------+----------+-------------+---------------------------+-----------+----------+
| St_roll_no | St_FName | St_LName    | Address                   | City      | phone_no |
+------------+----------+-------------+---------------------------+-----------+----------+
|    1750255 | Rohan    | Shrivastava | H.no-135, Near Pillikothi | Moradabad |        0 |
+------------+----------+-------------+---------------------------+-----------+----------+
1 row in set (0.00 sec)
```

Program Number: 17

Objective: Write a SQL query to create view and index.

Procedure:

Create View

```
mysql> CREATE VIEW RollNo AS
    -> select st_roll_no from students;
```

Query OK, 0 rows affected (0.19 sec)

Output

```
mysql> desc RollNo;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| st_roll_no | int(11) | YES  |     | NULL    |       |
+------------+---------+------+-----+---------+-------+
1 row in set (0.07 sec)

mysql> select * from rollno;
+------------+
| st_roll_no |
+------------+
|    1750255 |
+------------+
1 row in set (0.00 sec)
```

Create Index

```
mysql> CREATE INDEX index_stu
    ->   ON students (st_roll_no,st_FName);
```

Query OK, 1 row affected (0.23 sec)

Output

```
mysql> SHOW INDEXES FROM students;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| students |          1 | index_stu |            1 | St_roll_no  | A         |        NULL |     NULL | NULL   | YES  | BTREE      |         |               |
| students |          1 | index_stu |            2 | St_FName    | A         |        NULL |     NULL | NULL   | YES  | BTREE      |         |               |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
2 rows in set (0.06 sec)
```