→ Disable Intrupt.

⇒ Priority Intrupt.

→ Multi Programming.

Py

1994 - Opensource Scripting Language. - Syntax free, Indentation

⇒ web design.

⇒ Static.

Types of RDBAPI / SRDBMS

Sql → MySql → sqllite 3 → sql Alchemy → Postgre sql.

Syntax :-

MySql < Static / Dyn.

⇒ DDL Syntax :-

(1) Create database :.

```
Create database databasename;
```

Eg:-

```
Create database cs;
```

(2) Bring the database to use :-

```
Use databasename;
```

Eg:-

```
Use cs;
```

(3) Create a table.

```
Create table tblname ( field name data type (length)
```

Eg:-

```
Create table MCA (rollno variant (6) Primarykey, name String (20), ...)
```

(4) Drop table :-

```
Drop table. tblname;
```

Eg:-

```
Drop table MCA;
```

> DML Syntax :-

1) Delete :-

Delete * from tblo name ;
↳ All Records .

Eg:-
Delete * from MCA ;

2) Delete from tablename where condition.

Eg:-
Delet from MCA where mark<40 .

3) Insert :-

Insert Into tablename ("v1","v2","v3") ; → Static Insert

Eg:-
Insert Into MCA ("101", "x", "85");

5) update :-

update table tblname SET field = value where condition .

Eg:-
update table MCA Set mark =90 where rno = '102' ;

Steps for $\overset{M}{\text{DDL}}$ Python Program :-

Step 1 :- import the required Package .

Step 2 :- Create a connection Object using DBAPI connector .

Step 3 :- create a cursor object using connection object .

Step 4 :- excute the DML statement using cursor Object.

The Answer is reflected in DB .

ynamic :- Python Program for Inserting a Record dynamically.

yntax :-

INSERT INTO TABLE NAME (field 1, field 2, ...) VALUES

(?, ?, ?, ?) → Place holders.

Eg :-

```
import mysql
 conn = mysql.connect ("...../cs.db");
  cur = conn.cursor();
  rno = 15;
   name = abc;
 inscmd = " In/SERT INTO TYCA (rno name) VALUES(?,?)'
 cur.excute (inscmd, (rno, name)).
```

Python Program Insert a Record dynamically.

```
import mysql.
con = mysql.connect()
cur = con.cursor()
rno = input("Enter RollNo: ");
Name = input ( Enter Name : " );
inscmd = "INSERT INTO MCA (rno,name) VALUES (?,?
cur.execute (ins-cmd,(rno,name)).
con.commit()
cur.close()
con.close().
```

Inserting Multiple Records dynamically.

```
import mysql.as my...
con = mysql.connect(" ... "cs.db)
```

Syntax:

~~import mysql as my~~ . import db-api as aliasname.

```
import mysql as my
con = my.connect (".... /cs.db"):
cur = con.cursor().
inscmd = "INSERT INTO MCA (rno,Name ) VALUES (?,?)"
rec = [(1,'x') ,(2,'y'),(3,'z') ]

cur.executemany (ins-cmd,rec)
con.commit.
Cur.close()
Con.close().
```

Write a Pytho

Define a function :-

Syntax :-

```
def functionname :   } - defining a function.
    Statement

functionname ()      → calling a function.
```

Error handling :-

Syntax :-

```
try
    Statement.
    Statement
Except :
    Statement.
```

write a python to Insert, Delet, update records.
DML Statement.

```
import mysql as my
def connection():
    try:
        con = my.connect ("....,cs.d6").
    Except :
        Print ("Connection Error")
def Insert () :
    try :
        cur = con.cursor()
        inscmd = "INSERT INTO MCA (rno, name) VALUES(?,?)"
        rno = Input ("Enter rollno ");
        name= Input (Enter Name ")
        cur.execute (inscmd, (rno, name))
        con.commit ()
    Except :
        Print ("Insert Error")
```

```
def delete() :
 try: try :
   ~~del~~ " cur = con.cursor()
   rno = input ("Enter rollno"):
   del.cmd = " Delete FROM MCA WHERE r= 'r'        (or) ?
   cur.execute (del-cmd (r)) .                    '?'
  ~~cur~~. cur.close()
  Except :
   Print (" dele exception").

def update() :
  try :

     cur = con.cursor().
     rno  = input ("Enter rollno : ")
     ~~mark~~ = input ("Enter mark : ")
   up-cmd = "update table mCA set mark = 'mo' where rollno
                                                  'rno' "
   cur.execute ( upcmd)
    cur.close().
  Except :
    Print ("up cmd:").

connection ().
Insert()
delete()
update() .
```

Mozilla browser → Phpmyadmin → Create db → name of db → select → create table → name of table. → Field name

run → anaconda-navigator → spider

Con= redis.Redis().