

HANSRAJ MODEL SCHOOL

# **Computer Science**

**PROJECT FILE**

**(2020\_21)**

**CBSE Roll No. 26636499**

**Name: Rachit Tanwar**

# CERTIFICATE

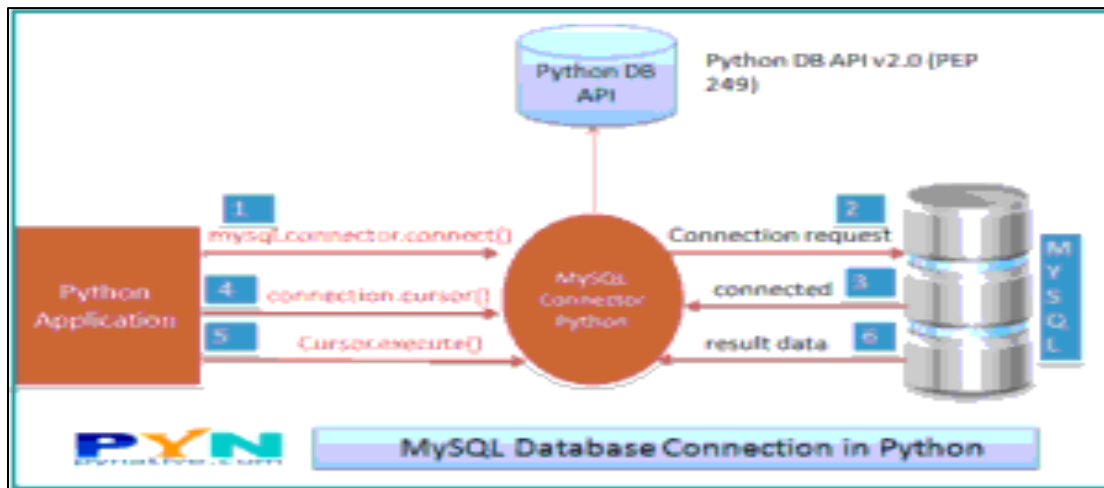
This is to certify **Rachit Tanwar** has worked under my supervision on the Computer Science Project titled **2048 Game** using Data Management concept to Interface Python with a SQL database. The project is completed to my satisfaction.

Mrs. **Navpreet Kaur Ma'am**  
(Computer Science Teacher)

## Client – Server Model

Client/ Server describes the relationship between two computer programs in which one program the client, makes a service request from another program, the server, which fulfills the request. Although programs within a single computer can use the client/server idea, it is a

more important idea in a network. In a network, the client server model provides a convenient way to interconnect programs that are distributed efficiently across different locations. Computer transactions using client/server model are very common.



## Front End and Back End

WHAT IS FRONT END?

It refers to the client-end which includes Python Application.

### **FRONT END SOFTWARE**

- Python version 3.7

### **WHAT IS BACK END?**

Back end development refers to the server side of an application and everything that communicates between the database and the browser.

### **BACK END SOFTWARE**

- DBMS – MySQL
- MySQL Connector version 8.0.15

# Database Connectivity

**Step 1:** import the MySQL Connector Python module in the program to be able to use the functions of this module to communicate with the MySQL database.

```
import mysql.connector
```

**try block** runs the code and test it for errors.

**Step 2:** Use `mysql.connector.connect()` function to connect the MySQL Database from Python

Example:

```
db = mysql.connector.connect (host='localhost',  
database='s1',
```

```
user='root', password='hrms')
```

`connect()` function takes 4 parameters namely host, user, password and database.

**host** - is the host name or IP address of the MySQL server. It can be either 'localhost' or '127.0.0.1'

**user** - is the user name used to authenticate with the MySQL server

**password** - The password to authenticate the user with the MySQL server.

**database** - is the database name to use

**Step 3:** You can create a cursor by executing the 'cursor' function of your database object.

Example: `cursor = db.cursor()`

**Step 4:** Executing queries is very simple in MySQL Python. All you need to do is take your cursor object and call the 'execute' function. The execute function requires one parameter, the query. If the query contains any substitutions then a second parameter, a tuple, containing the values to substitute must be given.

Use **execute()** function to execute a SQL query

Examples:

```
cursor.execute("select * from students order by marks")
cursor.execute("insert into student values(%s,%s,%s)",
(a,b,c) )
cursor.execute("update student set marks=%s where
rollno=%s", (m,r) )
cursor.execute("delete from student where rollno = %s",
(r,) )
```

On successful execution of a SELECT query, the **execute()** method returns a **ResultSet** object which contains all the rows.

**Step 5:** After you execute any *SELECT* statement, you will need to use one of these methods to obtain your results.

#### **Method 1: Fetch All-at-once**

use **cursor.fetchall()** method after SELECT query to fetch all the records from the result set

FOR SELECT statement, use for loop to iterate over all the records present under **resultset** object and print them one by one.

#### **Method 2: Fetch-One-at-time**

Use **cursor.fetchone()** method after SELECT query to fetch one record in the result set.

**Step 6:** use **db.commit ()** function to make the changes of **INSERT, UPDATE, DELETE** permanent in the database table

**Step 7:** Use `cursor.close()` and `db.close()` to close the cursor object and connection object respectively.

**except block** handles the error in try block

**Step 8:** Use `db.rollback()` to undo the changes made by INSERT/ UPDATE/ DELETE statements and display a suitable error message.

## **INPUT**

```
import random
import copy
import mysql.connector

#Creating mysql connection
mydb = mysql.connector.connect( host = "localhost",
user = "root",password = "1234",database = "2048game")
mycursor = mydb.cursor()
#Create the board size variable
boardsize = 4
# This function will print out the current board
def display():
    # Find out which value is largest
```

```

largest = board[0][0]
for row in board:
    for element in row:
        if element > largest:
            largest = element
# Set the max number of spaces needed in the length
of the largest value
numspaces = len(str(largest))
for row in board:
    currRow = "|"
    for element in row:
        # if the current element is 0, add a space
        if element == 0 :
            currRow += " " * numspaces + "|"
        # if not, we should add the value
        else:
            currRow += (" " * (numspaces -
len(str(element)))) + str(element) + "|"
        # print the generated row.
        print(currRow)
    print()

#This function merges one row left
def mergeonerowLEFT(row):
    #Move everything as far to the left as possible
    for j in range(boardsize - 1):
        for i in range(boardsize - 1, 0 , -1):
            #Test if there is an empty space,move over
            if so
                if row[i - 1] == 0:
                    row[i - 1] = row[i]
                    row[i] = 0
    '''Merge everything to the left'''
    for i in range(boardsize - 1):

```



```

        '''Test if the current value is identical to
the one next to it'''
        if row[i] == row[i + 1]:
            row[i] *= 2
            row[i + 1] = 0
        '''Move everything to the left again'''
    for i in range(boardsize - 1, 0 , -1):
        if row[i - 1] == 0:
            row[i - 1] = row[i]
            row[i] = 0
    return row

''' This function merges the whole board to the left'''
def merge_LEFT(currentboard):
    # Merge every row in the board left
    for i in range(boardsize):
        currentboard[i] =
mergeonerowLEFT(currentboard[i])
    return currentboard

# This function reverses the order of one row
def reverse(row):
    ''' Add all elements of the row to a new list, in
reverse order'''
    new = []
    for i in range(boardsize -1 , -1, -1):
        new.append(row[i])
    return new

# This function merges the whole board right
def merge_RIGHT(currentboard):
    ''' Look at every row in the board'''
    for i in range(boardsize):
        # Reverse the row, merge to the left , then
reverse back

```

```

        currentboard[i] = reverse(currentboard[i])
        currentboard[i] =
mergeonerowLEFT(currentboard[i])
        currentboard[i] = reverse(currentboard[i])
    return currentboard

# This function transposes the whole board
def transpose(currentboard):
    for j in range(boardsize):
        for i in range(j , boardsize):
            if not i == j :
                temp = currentboard[j][i]
                currentboard[j][i] = currentboard[i][j]
                currentboard[i][j] = temp
    return currentboard

''' This function merges the whole board up '''
def merge_UP(currentboard):
    ''' Transposes the whole board, merges it all left,
    then transpose it back'''

    currentboard = transpose(currentboard)
    currentboard = merge_LEFT(currentboard)
    currentboard = transpose(currentboard)

    return currentboard

# This function merges the whole board down
def merge_DOWN(currentboard):
    ''' Transposes the whole board, merges it all
    right, then transpose it back'''

    currentboard = transpose(currentboard)
    currenboard = merge_RIGHT(currentboard)
    currentboard = transpose(currentboard)

```

```

    return currentboard

# This function picks a new value for the board
def picknewvalue():
    if random.randint(1 , 8) == 1:
        return 4
    else:
        return 2

# This function adds a value to the board in one of the
empty space
def addnewvalue():
    global score
    score += 1
    row_num = random.randint(0, boardsize - 1)
    col_num = random.randint(0, boardsize - 1)

    while not board[row_num][col_num] == 0:
        row_num = random.randint(0, boardsize - 1)
        col_num = random.randint(0 , boardsize - 1)

    board[row_num][col_num] = picknewvalue()

def won():
    for row in board:
        if 2048 in row:
            return True
    return False

def nomoves():
    # Create two copies of the board()
    tempboard1 = copy.deepcopy(board)
    tempboard2 = copy.deepcopy(board)

```

```

tempboard1 = merge_DOWN(tempboard1)
if tempboard1 == tempboard2:
    tempboard1 = merge_UP(tempboard1)
    if tempboard1 == tempboard2:
        tempboard1 = merge_LEFT(tempboard1)
        if tempboard1 == tempboard2:
            tempboard1 = merge_RIGHT(tempboard1)
            if tempboard1 == tempboard2:
                return True
return False

# Create a blank board
board = []
score = 0
for i in range(boardsize):
    row = []
    for j in range(boardsize):
        row.append(0)
    board.append(row)

''' Fill two spots with random values, to start the
game'''
num_needed = 2
while num_needed > 0 :
    row_num = random.randint(0 , boardsize - 1)
    col_num = random.randint(0 , boardsize - 1)

    if board[row_num][col_num] == 0:
        board[row_num][col_num] = picknewvalue()
        num_needed -= 1

# STARTING MENU
print('Welcome to 2048! Your goal is to combine values
to get the number 2048, by merging the board in

```

different directions. Everytime, you will need to "s" to merge down, "w" to merge up, "a" to merge left and "d" to merge right. \n\n Here is the starting board::')

```
print("Press 1 for New Player")
print("Press 2 for Existing player")
choice = int(input())
Name = input('Enter Your Full Name : ')
if choice == 1:
    SQL = 'Insert into players (name) values(%s)'
    VAL = (Name.title(),)
    mycursor.execute(SQL,VAL)
    mydb.commit()
    print(mycursor.rowcount,"User Registered")
    print("Press 1 for playing")
    option = int(input())
    if option == 1:
        display()
else:
    print("Press 1 for playing")
    print("Press 2 for seeing your highest score")
    print("Press 3 for seeing the winners")
    option = int(input())

    if option == 1:
        display()
    elif option == 2:

        SQL = 'Select max(score) from players where
name like %s '
        VAL = (Name.title(),)
        mycursor.execute(SQL,VAL)
        result = mycursor.fetchone()
        print('YOUR HIGHEST SCORE:')
        print(result)
```

```
        print("Start new round by pressing any key from  
wasd")
```

```
    elif option == 3:  
        status = "Win"  
        SQL = 'Select * from players where status like  
%s '  
        VAL = (status,)   
        mycursor.execute(SQL,VAL)  
        result = mycursor.fetchall()  
        if result == []:  
            print("There is no winner yet!")  
        else:  
            print('WINNERS:')  
            for i in result:  
                print(i)  
        print("Start new round by pressing any key from  
wasd")
```

```
game_over = False  
# Repeat asking the user for new moves while the game  
isn't over  
while not game_over :  
    print('Your score right now = ', score)  
    move = input("Which way you want to merge?")  
  
    # Assume they entered a valid input  
    valid_input = True  
  
    # Create a copy of the board  
    tempboard = copy.deepcopy(board)  
  
    ''' Figure out which way the person wants to merge  
and use the  
correct function'''
```

```

if move == 'w':
    board = merge_UP(board)
elif move == 's':
    board = merge_DOWN(board)
elif move == 'd':
    board = merge_RIGHT(board)
elif move == 'a':
    board = merge_LEFT(board)
else:
    valid_input = False

if not valid_input:
    print('Your input was not valid, please try
again')
else:
    if board == tempboard:
        print('Try a different direction!')
        end = input("Want to continue or exit(y for
exit / any key for continue)")
        if end == "y":
            game_over = True
            SQL = "Insert into players (name,score)
values(%s,%s)"
            VAL = (Name.title(),score)
            mycursor.execute(SQL,VAL)
            print('Record updated')
            print("Thank you so much for playing")
            mydb.commit()
            print("Your final score is ", score)
            print("Press 1 for playing")
            print("Press 2 for seeing your highest
score")

            print("Press 3 for seeing the winners")
            option = int(input())

```

```

        if option == 1:
            display()
            game_over = False
        elif option == 2:
            SQL = 'Select max(score) from
players where name like %s '
            VAL = (Name.title(),)
            mycursor.execute(SQL,VAL)
            result = mycursor.fetchone()
            print('YOUR HIGHEST SCORE:')
            print(result)

        elif option == 3:
            status = "Win"
            SQL = 'Select * from players where
status like %s '
            VAL = (status,)
            mycursor.execute(SQL,VAL)
            result = mycursor.fetchall()
            if result == []:
                print("There is no winner
yet!")
            else:
                print('WINNERS:')
                for i in result:
                    print(i)

    else:
        # Test if user has won
        if won():
            display()
            status = "Win"
            SQL = "Insert into players
(name,score,status) values(%s,%s,%s) "

```



```

        VAL = (Name.title(),score,status)
        mycursor.execute(SQL,VAL)
        mydb.commit()
        print("YOU WON!!!!")
        game_over = True

    else:
        addnewvalue()
        display()
        # Test if user lost
        if nomoves():
            status = "Lose"
            print("SORRY! You have no more
possible move, you lose!")
            SQL = "insert into players
(name,score,status) values(%s,%s,%s)"
            VAL = (Name.title(),score,status)
            mycursor.execute(SQL,VAL)
            mydb.commit()
            print("Your final score is ",
score)

            SQL = "Update players set score =
%s where name = %s"
            VAL = (score,Name.title())
            mycursor.execute(SQL,VAL)
            mydb.commit()
            game_over = True

```

## **OUTPUT**

## PLAYING VIEW

```
>>>
```

```
===== RESTART: D:\Rachit\Class 12\Computer\CS School  
Project\CS PROJECT.py =====
```

```
Welcome to 2048! Your goal is to combine values to get  
the number 2048, by merging the board in different  
directions. Everytime, you will need to "s" to merge  
down, "w" to merge up, "a" to merge left and " d" to  
merge right.
```

```
Here is the starting board::
```

```
Press 1 for New Player
```

```
Press 2 for Existing player
```

```
1
```

```
Enter Your Full Name : Rachit Tanwar
```

```
1 User Registered
```

```
Press 1 for playing
```

```
1
```

```
| | | | |  
| |2| | |  
| | | | |  
|2| | | |
```

```
Your score right now = 0
```

```
Which way you want to merge?a
```

```
| | | | |  
|2| | | |  
| | |2| |  
|2| | | |
```

```
Your score right now = 1
```

Which way you want to merge?d

```
| | | | |
|2| | |2|
| | | |2|
| | | |2|
```

Your score right now = 2

Which way you want to merge?s

```
| | | | |
| | | | |
| | | |2|
|2| |2|4|
```

Your score right now = 3

Which way you want to merge?d

```
| |2| | |
| | | | |
| | | |2|
| | |4|4|
```

Your score right now = 4

Which way you want to merge?w

```
| |2|4|2|
| | | |4|
| | |2| |
| | | | |
```

Your score right now = 5

Which way you want to merge?a

```
|2|4|2| |
|4| | | |
|2| | | |
| | |2| |
```

Your score right now = 6

Which way you want to merge?s

		2	
2			
4			
2	4	4	

Your score right now = 7

Which way you want to merge?d

			2
			2
2			4
		2	8

Your score right now = 8

Which way you want to merge?w

2		2	4
2			4
			8

Your score right now = 9

Which way you want to merge?a

4	4		
2	4		
	2		
8			

Your score right now = 10

## **LOST VIEW**

Which way you want to merge?w

	4	8	2	32
	128	64	8	2
	2	8	256	4
	4	16	4	2

SORRY! You have no more possible move, you lose!  
Your final score is 238

## **BIBLIOGRAPHY**

- 1) Schools notes
- 2) [www.w3schools.com](http://www.w3schools.com)
- 3) [www.stackoverflow.com](http://www.stackoverflow.com)
- 4) [www.google.com](http://www.google.com)