

Name:Kartikay Agrawal

Roll No:2148064

▼ LAB TEST-2

PART A: Write a python code to perform the following task using regular expression concepts:

2.5x4=10 Marks

- i) Find the dates which accept and match the date formats either dd/mm/yy or yy/mm/dd
- ii) Not to accept 30th or 31st in February month
- iii) Accept and validate the emails ending only with .in or .net (Give an input of any 10 list of emails).
- iv) Find all the occurrences having 'in' in the list of emails

```
import re
# for validating a Date
regex = '^([0]?[1-9]|[1|2][0-9]|[3][0|1])[./-]([0]?[1-9]|[1][0-2])[./-]([0-9]{2})$'

def check(date):
    if(re.search(regex,date)):
        print("Valid date")
    else:
        print("Invalid date")

# Enter the date
date="32/04/2010"
check(date)

date="24/08/20"
check(date)

Invalid date
Valid date

# for validating a Date in february
regex = '^([0]?[1-9]|[1|2][0-9])[./-]([0][2])[./-]([0-9]{2})$'

def check(date):
    if(re.search(regex,date)):
        print("Valid date")
    else:
        print("Invalid date")

# Enter the date
date="30/02/11"
```

```
check(date)
```

```
date="24/02/11"
```

```
check(date)
```

```
Invalid date
```

```
Valid date
```

```
# for validating an Email
```

```
regex = '^[a-z0-9]+[\.\_]?[a-z0-9]+[@]\w+[.][inet]{2,3}$'
```

```
def check(email):
```

```
    if(re.search(regex,email)):
```

```
        print("Valid Email")
```

```
    else:
```

```
        print("Invalid Email")
```

```
# Enter the email
```

```
email="anubinrathi326@yahoo.in"
```

```
check(email)
```

```
email="jaka234@gmail.com"
```

```
check(email)
```

```
email="anarkali326.net"
```

```
check(email)
```

```
Valid Email
```

```
Invalid Email
```

```
Invalid Email
```

```
# for validating an Email (only in)
```

```
regex = '^[a-z0-9]+[\.\_]?[a-z0-9]+[@]\w+[.][in]{2}$'
```

```
def check(email):
```

```
    if(re.search(regex,email)):
```

```
        print("Valid Email")
```

```
    else:
```

```
        print("Invalid Email")
```

```
# Enter the email
```

```
email="anubinrathi326@yahoo.in"
```

```
check(email)
```

```
email="jaka234@gmail.com"
```

```
check(email)
```

```
email="anarkali326@gmail.net"
```

```
check(email)
```

```
Valid Email
```

```
Invalid Email
```

```
Invalid Email
```

PART B: Create a table with following fields and create at least 5 rows and perform the following tasks: 10 Marks

Create a Python Program to create a student database and execute create, delete, update and select commands of SQL (5x1=5).

- i. Add a new column called total which holds the sum of all 3 subjects. 1Mark
- ii. Select the instances where the total marks are greater than >400 and gender is Female. 1Mark
- iii. Find all the subjects that has 'in' in it. 1Mark
- iv. Demonstrate the usage of group by and having clause. 1Mark
- v. Sort the total in descending order. 1Mark

```
import sqlite3
conn = sqlite3.connect('student.db')
```

```
conn.execute('''CREATE TABLE STUDENTS (SNAME TEXT NOT NULL,REGNUM NUMERIC(10), G
print ("Table created successfully")
```

```
↳ -----
OperationalError                                Traceback (most recent call last)
<ipython-input-5-81cbeb355d4d> in <module>()
      2 conn = sqlite3.connect('student.db')
      3
----> 4 conn.execute('''CREATE TABLE STUDENTS (SNAME TEXT NOT NULL,REGNUM
NUMERIC(10), GENDER VARCHAR(10),PROGRAM VARCHAR(20),SUB1 VARCHAR(30),SUB2
VARCHAR(30),SUB3 VARCHAR(30),TEACHER VARCHAR(30));''')
      5
      6 print ("Table created successfully")
```

OperationalError: table STUDENTS already exists

SEARCH STACK OVERFLOW

```
conn.execute("INSERT INTO STUDENTS VALUES ('MANISH', 25, 'MALE', 'MBA', 'BUSINESS',
conn.execute("INSERT INTO STUDENTS VALUES ('SHIVA', 26, 'MALE', 'MSC', 'PHYSICS', 'C
conn.execute("INSERT INTO STUDENTS VALUES ('SURESH', 51, 'MALE', 'MA', 'JOURNALISM'
conn.execute("INSERT INTO STUDENTS VALUES ('DIKSHA', 55, 'FEMALE', 'MA', 'FRENCH', '
conn.execute("INSERT INTO STUDENTS VALUES ('GAURAV', 65, 'MALE', 'MTECH', 'ENGLISH'
```

```
conn.commit()
```

#1

```
conn.execute("ALTER TABLE STUDENTS ADD TOTAL NUMERIC(20)");
```

```
conn.execute("UPDATE STUDENTS SET TOTAL = 345 WHERE REGNUM=25; ");
```

```
conn.execute("UPDATE STUDENTS SET TOTAL = 450 WHERE REGNUM=26; ");
conn.execute("UPDATE STUDENTS SET TOTAL = 410 WHERE REGNUM=51; ");
conn.execute("UPDATE STUDENTS SET TOTAL = 433 WHERE REGNUM=55; ");
conn.execute("UPDATE STUDENTS SET TOTAL = 355 WHERE REGNUM=65; ");
```

#2

```
cursor = conn.execute("SELECT * FROM STUDENTS WHERE TOTAL>400 AND GENDER='FEMALE'")
```

#3

```
def Like_Data():
```

```
# Connecting to sqlite
```

```
    conn = sqlite3.connect('student.db')
```

```
#Creating a cursor object using
```

```
    cursor = conn.cursor()
```

```
    cursor.execute(''SELECT * FROM STUDENTS WHERE SUB1 LIKE '%IN%';'')
```

```
    cursor.execute(''SELECT * FROM STUDENTS WHERE SUB2 LIKE '%IN%';'')
```

```
    cursor.execute(''SELECT * FROM STUDENTS WHERE SUB3 LIKE '%IN%';'')
```

```
    result = cursor.fetchall()
```

```
    for i in result:
```

```
        print(i)
```

```
# Commit your changes in the database
```

```
    conn.commit()
```

```
# Closing the connection
```

```
    conn.close()
```

```
Like_Data()
```

#4

```
cursor=conn.execute(''SELECT COUNT(REGNUM), PROGRAM, TOTAL FROM STUDENTS GROUP BY PROC
```

```
result = cursor.fetchall()
```

```
for i in result:
```

```
    print(i)
```

#5

```
cursor = conn.execute("SELECT * FROM STUDENTS ORDER BY TOTAL DESC")
```

 0s completed at 12:58 PM  