18CSC207J-Advance Programming Practice - Structured Programming – Lab Programs

Lab 5 Declarative Programming Paradigm

Name :- Puneet Sharma
Reg. No. :- RA1911003010331
Class :-CSE F1

Graded:

1. Create the below table and execute the insert, update and the below select statements.

i) Write a query to display the total number of recipes available with the description
“Chinese”
ii) Write a query to display the id, name of the recipes with chef_id 'BL000002'.
iii) Write a query to display the description of the recipes whose name begins with 'P'.

CODE:-

```
import sqlite3
conn = sqlite3.connect('recipesdb')
curs = conn.cursor()
curs.execute('drop table recipes')
cmd="create table recipes (id int(11) ,name varcar(400), description text, category_id int(11), chef_id int (255), created datetime)"
conn.commit()
curs.execute(cmd)
curs.execute('insert into recipes values(1,"Puneet","good",111,000001,"20210309 23:53:12 PM")')
curs.execute('insert into recipes values(2,"P1","good",222,000002,"20210306 23:53:12 PM")')
curs.execute('insert into recipes values(2,"A2","good",333,000003,"20210301 23:53:12 PM")')
curs.execute('insert into recipes values(3,"B4","Chinese",333,000003,"20210301 23:53:12 PM")')

print("Before update")
curs.execute('select * from recipes')
for row in curs.fetchall():
    print(row)
print("\nAfter update")
curs.execute('update recipes set name="D3" where id = 2')
```

```
curs.execute('select * from recipes')
for row in curs.fetchall():
    print(row)
print("\n Part 1")
p2=curs.execute('select id, name from recipes where description="Chinese"')
for r in p2:
    print(r)
print("\n Part 3")
p3=curs.execute('select * from recipes where name like "P%"')
for row in p3:
    print(row)
```

```
#Question 1

import sqlite3
conn = sqlite3.connect('recipesdb')
curs = conn.cursor()
curs.execute('drop table recipes')
cmd="create table recipes (id int(11) ,name varcar(400), description text, category_id int(11),\
chef_id int (255), created datetime)"
conn.commit()
curs.execute(cmd)
curs.execute('insert into recipes values(1,"Puneet","good",111,000001,"20210309 23:53:12 PM")')
curs.execute('insert into recipes values(2,"P1","good",222,000002,"20210306 23:53:12 PM")')
curs.execute('insert into recipes values(2,"A2","good",333,000003,"20210301 23:53:12 PM")')
curs.execute('insert into recipes values(3,"B4","Chinese",333,000003,"20210301 23:53:12 PM")')

print("Before update")
curs.execute('select * from recipes')
for row in curs.fetchall():
    print(row)
print("\nAfter update")
curs.execute('update recipes set name="D3" where id = 2')
curs.execute('select * from recipes')
for row in curs.fetchall():
    print(row)
print("\n Part 1")
p2=curs.execute('select id, name from recipes where description="Chinese"')
for r in p2:
    print(r)
print("\n Part 3")
p3=curs.execute('select * from recipes where name like "P%"')
for row in p3:
    print(row)
```

Output:

```
Before update
(1, 'Puneet', 'good', 111, 1, '20210309 23:53:12 PM')
(2, 'P1', 'good', 222, 2, '20210306 23:53:12 PM')
(2, 'A2', 'good', 333, 3, '20210301 23:53:12 PM')
(3, 'B4', 'Chinese', 333, 3, '20210301 23:53:12 PM')

After update
(1, 'Puneet', 'good', 111, 1, '20210309 23:53:12 PM')
(2, 'D3', 'good', 222, 2, '20210306 23:53:12 PM')
(2, 'D3', 'good', 333, 3, '20210301 23:53:12 PM')
(3, 'B4', 'Chinese', 333, 3, '20210301 23:53:12 PM')

 Part 1
(3, 'B4')

 Part 3
(1, 'Puneet', 'good', 111, 1, '20210309 23:53:12 PM')
```

2. Create a table movie of the below structure and assume data types.Movie_ID, Movie_Name, Genre, Language, Rating ,Do the following queries
a. Update the movies rating by 10% and display it
b. Delete the movies with movie_id 102
c. Select movies whose rating is more than 3.

CODE:

```
import sqlite3
conn=sqlite3.connect('lab5')
curs=conn.cursor()
curs.execute('drop table movie')
cmd="create table movie(Movie_ID int(200) ,Movie_name varchar(100) , Genre
varchar(100) , Language varchar(50) , Rating int(10))"
curs.execute(cmd)
curs.execute('insert into movie values(1,"Ironman","SiFi","English",5)')
curs.execute('insert into movie values(2,"Avengers","SiFi","English",5)')
curs.execute('insert into movie values(10,"Spiderman","SiFi","English",5)')
curs.execute('insert into movie values(102,"MadMax","SiFi","English",2)')
curs.execute('select * from movie')
for row in curs.fetchall():
    print(row)
print("\n Part 1")
curs.execute('update movie set Rating=Rating+(Rating*0.10)')
curs.execute('select * from movie')
for row in curs.fetchall():
    print(row)
print("\n Part 2")
p2=curs.execute('select * from movie where Movie_ID=102')
for r in p2:
    print(r)
print("\n Part 3")
p3=curs.execute('select * from movie where Rating>3')
for r in p3:
    print(r)
```

```
#Question 2

import sqlite3
conn=sqlite3.connect('lab5')
curs=conn.cursor()
curs.execute('drop table movie')
cmd="create table movie(Movie_ID int(200) ,Movie_name varchar(100) , Genre varchar(100) , Language varchar(50)\
, Rating int(10))"
curs.execute(cmd)
curs.execute('insert into movie values(1,"Ironman","SiFi","English",5)')
curs.execute('insert into movie values(2,"Avengers","SiFi","English",5)')
curs.execute('insert into movie values(10,"Spiderman","SiFi","English",5)')
curs.execute('insert into movie values(102,"MadMax","SiFi","English",2)')
curs.execute('select * from movie')
for row in curs.fetchall():
    print(row)
print("\n Part 1")
curs.execute('update movie set Rating=Rating+(Rating*0.10)')
curs.execute('select * from movie')
for row in curs.fetchall():
    print(row)
print("\n Part 2")
p2=curs.execute('select * from movie where Movie_ID=102')
for r in p2:
    print(r)
print("\n Part 3")
p3=curs.execute('select * from movie where Rating>3')
for r in p3:
    print(r)
```

Output:

```
(1, 'Ironman', 'SiFi', 'English', 5)
(2, 'Avengers', 'SiFi', 'English', 5)
(10, 'Spiderman', 'SiFi', 'English', 5)
(102, 'MadMax', 'SiFi', 'English', 2)

 Part 1
(1, 'Ironman', 'SiFi', 'English', 5.5)
(2, 'Avengers', 'SiFi', 'English', 5.5)
(10, 'Spiderman', 'SiFi', 'English', 5.5)
(102, 'MadMax', 'SiFi', 'English', 2.2)

 Part 2
(102, 'MadMax', 'SiFi', 'English', 2.2)

 Part 3
(1, 'Ironman', 'SiFi', 'English', 5.5)
(2, 'Avengers', 'SiFi', 'English', 5.5)
(10, 'Spiderman', 'SiFi', 'English', 5.5)
```

3. Create a course database with the following fields Product(ID, Prod_name, Supplier_id,Unit_price,Package,OrderID),OrderItem(ID,Order_id,Product_id,Unit_price,
Quantity) using Foreign key
d. Display the total quantity of every product in the stock
e. Sort the Unit_price based on the supplier_id
f. Display the Product_name along with order_id and supplier_id

CODE:

```
import sqlite3
conn=sqlite3.connect('course')
curs=conn.cursor()
curs.execute('drop table product')
t1="create table product (ID int(5) , Prod_name varchar(20) , Suppliers_id int(10) ,
Unit_price int(10),\
Package varchar(20) , Order_id int(10) ,PRIMARY KEY(ID))"
curs.execute(t1)
rowst1 =[[2,"Shirt",5,50,"prime",6],[3,"shoes",8,60,"normal",7],[6,"pants",8,90,"prime",9]]
curs.executemany("insert into product values(?,?,?,?,?,?)",rowst1)
conn.commit()
curs.execute('drop table OrderItem')
t2="create table OrderItem (ID INT(5),Order_id int(10), Prod_id int(10),Unit_price
REAL,Quantity int(10),\
FOREIGN KEY(Prod_id) REFERENCES product(Prod_id))"
curs.execute(t2)
curs.executemany("INSERT INTO OrderItem
VALUES(?,?,?,?,?)",[(1,5,2,20,8),(2,7,3,30,6),(3,3,6,28.90,5)])
print("\n Part 1")
p1=curs.execute('select Prod_id,Quantity from OrderItem ')
for r in p1:
    print(r)
print("\n Part 2")
p2=curs.execute("select Unit_price,Suppliers_id FROM product ORDER BY Unit_price")
for r in p2:
    print(r)
print("\n Part 3")
p3=curs.execute("select Prod_name,Order_id,Suppliers_id FROM product")
for r  in p3:
    print(r)
```

```python
#Question 3

import sqlite3
conn=sqlite3.connect('course')
curs=conn.cursor()
curs.execute('drop table product')
t1="create table product (ID int(5) , Prod_name varchar(20) , Suppliers_id int(10) , Unit_price int(10),\
Package varchar(20) , Order_id int(10) ,PRIMARY KEY(ID))"
curs.execute(t1)
rowst1 =[[2,"Shirt",5,50,"prime",6],[3,"shoes",8,60,"normal",7],[6,"pants",8,90,"prime",9]]
curs.executemany("insert into product values(?,?,?,?,?,?)",rowst1)
conn.commit()
curs.execute('drop table OrderItem')
t2="create table OrderItem (ID INT(5),Order_id int(10), Prod_id int(10),Unit_price REAL,Quantity int(10),\
FOREIGN KEY(Prod_id) REFERENCES product(Prod_id))"
curs.execute(t2)
curs.executemany("INSERT INTO OrderItem VALUES(?,?,?,?,?)",[(1,5,2,20,8),(2,7,3,30,6),(3,3,6,28.90,5)])
print("\n Part 1")
p1=curs.execute('select Prod_id,Quantity from OrderItem ')
for r in p1:
    print(r)
print("\n Part 2")
p2=curs.execute("select Unit_price,Suppliers_id FROM product ORDER BY Unit_price")
for r in p2:
    print(r)
print("\n Part 3")
p3=curs.execute("select Prod_name,Order_id,Suppliers_id FROM product")
for r  in p3:
    print(r)

```

Output:

```
 Part 1
(2, 8)
(3, 6)
(6, 5)

 Part 2
(50, 5)
(60, 8)
(90, 8)

 Part 3
('Shirt', 6, 5)
('shoes', 7, 8)
('pants', 9, 8)
```

4. Write a SQL lite3 statement to create a table named as job including columns job_id,job_title,Min-salary,Max_salary.job_id column does not contain any duplicate
value at the time of insertion

CODE:

```
import sqlite3
conn=sqlite3.connect("lab5")
curs=conn.cursor()
curs.execute("DROP table job")
table="create table job (job_id int(10) PRIMARY KEY, job_title varchar(20) , min_salary int(10) , max_salary int(10))"
curs.execute(table)
rows=[[1,'Manager',1000,50000],[2,'Assisistent',1000,30000]]
curs.executemany("insert into job values(?,?,?,?)",rows)
a=curs.execute("select * from job")
for r in a:
    print(r)
curs.executemany("insert into job values(?,?,?,?)",rows)
```

```
#Question 4

import sqlite3
conn=sqlite3.connect("lab5")
curs=conn.cursor()
curs.execute("DROP table job")
table="create table job (job_id int(10) PRIMARY KEY, job_title varchar(20) , min_salary int(10) , max_salary int(10))"
curs.execute(table)
rows=[[1,'Manager',1000,50000],[2,'Assisistent',1000,30000]]
curs.executemany("insert into job values(?,?,?,?)",rows)
a=curs.execute("select * from job")
for r in a:
    print(r)
curs.executemany("insert into job values(?,?,?,?)",rows)
```

Output:

```
(1, 'Manager', 1000, 50000)
(2, 'Assisistent', 1000, 30000)

---------------------------------------------------------------------------
IntegrityError                            Traceback (most recent call last)
<ipython-input-3-814876b685fa> in <module>
     12 for r in a:
     13     print(r)
---> 14 curs.executemany("insert into job values(?,?,?,?)",rows)

IntegrityError: UNIQUE constraint failed: job.job_id
```

5. Write a SQL lite3 statement to create a table names as job_history including columns
employee_id, start_date, end_date, job_id and department_id and make sure that, the
employee_id column does not contain any duplicate value at the time of insertion and the
foreign key column job_id contain only those values which are exists in the jobs table.

CODE:

```
import sqlite3
conn=sqlite3.connect("lab5")
curs=conn.cursor()
curs.execute("drop table job_history")
curs.execute("drop table job")
t1="create table job (job_id int(10) PRIMARY KEY, job_title varchar(20) , min_salary int(10) , max_salary int(10))"
curs.execute(t1)
curs.executemany("INSERT INTO job VALUES(?,?,?,?)",[[1,'Manager',1000,50000],[2,'Assisistent',1000,30000]])
conn.commit()
t2="create table job_history (employee_id int(5) PRIMARY KEY,start_date varcar(5),end_date varcar(5),job_id int(5),FOREIGN KEY('job_id') references job('job_id'))"
curs.execute(t2)
curs.executemany("INSERT INTO job_history VALUES(?,?,?,?)",[ (1,'2 feb 2015','6 june 2018',1),(2,'8 dec 2013','3 jan 2021',2)])
conn.commit()
p=curs.execute("SELECT * FROM job_history")
for r in p:
    print(r)
```

```
#Question 5

import sqlite3
conn=sqlite3.connect("lab5")
curs=conn.cursor()
curs.execute("drop table job_history")
curs.execute("drop table job")
t1="create table job (job_id int(10) PRIMARY KEY, job_title varchar(20) , min_salary int(10) , max_salary int(10))"
curs.execute(t1)
curs.executemany("INSERT INTO job VALUES(?,?,?,?)",[[1,'Manager',1000,50000],[2,'Assisistent',1000,30000]])
conn.commit()
t2="create table job_history (employee_id int(5) PRIMARY KEY,start_date varcar(5),end_date varcar(5),job_id int(5),\
FOREIGN KEY('job_id') references job('job_id'))"
curs.execute(t2)
curs.executemany("INSERT INTO job_history VALUES(?,?,?,?)"\
                 ,[ (1,'2 feb 2015','6 june 2018',1),(2,'8 dec 2013','3 jan 2021',2)])
conn.commit()
p=curs.execute("SELECT * FROM job_history")
for r in p:
    print(r)
```

Output:

```
(1, '2 feb 2015', '6 june 2018', 1)
(2, '8 dec 2013', '3 jan 2021', 2)
```