

## Console application source Code

```
# -*- coding: utf-8 -*-
```

```
''''
```

```
Created on Sun Nov 17 14:33:03 2019
```

```
@author: This-PC
```

```
''''
```

```
import psycopg2
```

```
from tabulate import tabulate
```

```
query=[]
```

```
query.append("with r1 as (select itemcode,(qty*avg_cp) as avg_spent from ( select  
itemcode,avg(cost_price) as avg_cp,sum(qty) as qty from items natural join supply_record where  
date>='2019-10-01' and to_date('2019-10-01','YYYY-MM-DD')+items.sellout_period>=date group by  
itemcode)as r), r2 as (select itemcode,sum(qty*purchaseprice) as recv from bill natural join  
bill_details natural join items where bill_date>='2019-10-01' and to_date('2019-10-01','YYYY-MM-  
DD')+items.sellout_period>=bill_date group by itemcode) select itemcode,recv-avg_spent as profit  
from r1 natural join r2;")
```

```
query.append("select sum(recv-spent) as profit from (( select  
p.itemcode,sum(p.qty)*avg(p.cost_price) as spent from (select * from supply_record as s where  
s.date>='2019-10-10' and s.date<='2019-11-10') as p group by p.itemcode)as e natural join (select  
v.itemcode,sum(v.qty*v.purchaseprice) as recv from (select * from bill_details as w natural join bill  
as b where b.bill_date>='2019-10-10' and b.bill_date<='2019-11-10') as v group by v.itemcode)as  
t)as q;")
```

```
query.append("with r2 as (with r1 as (select itemcode,sum(qty) as numberofitemsold from  
bill_details as b natural join bill as bb where bb.bill_date>='2019-10-01' and bb.bill_date<='2019-10-  
31' group by itemcode) select * from r1 natural join (select max(numberofitemsold) as  
numberofitemsold from r1) as e) (select itemcode,productname,numberofitemsold from r2 natural  
join packed_food_description) union (select itemcode,productname,numberofitemsold from r2  
natural join clothes_description) union (select itemcode,productname,numberofitemsold from r2  
natural join personal_care_description)")
```

```
query.append("with r1 as (select serviced_by,count(serviced_by) as mostresolved from complain  
natural join bill where bill_date>='2019-10-01' and bill_date<='2019-10-31' and status='resolved'  
group by serviced_by) select serviced_by,name,mostresolved from (r1 join employee on  
r1.serviced_by=employee.ssn) natural join (select max(mostresolved) as mostresolved from r1) as  
e;")
```

```
query.append("with r1 as (select brandname , sum(numberofitemsold) as brandcount from (select *  
from (select * from product natural join ((select itemcode,productname from  
packed_food_description) union (select itemcode,productname from clothes_description) union
```

(select itemcode,productname from personal\_care\_description)) as p ) as v natural join (select itemcode,sum(qty) as numberofitemsold from bill\_details as b natural join bill as bb where bb.bill\_date>='2019-10-01' and bb.bill\_date<='2019-10-31' group by itemcode) as f) as d group by brandname) select \* from r1 natural join (select max(brandcount) as brandcount from r1) as e")

query.append("select \* from packed\_food\_description natural join (with r1 as (select itemcode,count(itemcode) as no\_of\_complains from packed\_food\_description natural join bill\_details natural join complain group by itemcode) select \* from r1 natural join (select max(no\_of\_complains) as no\_of\_complains from r1) as q) as c")

query.append("with r1 as (select discount\_applied,count(discount\_applied) as mostusedcode from bill\_details natural join bill where bill\_date>='2019-10-01' and bill\_date<='2019-10-30' and not discount\_applied is null group by discount\_applied) select discount\_applied,mostusedcode from (select max(mostusedcode)as mostusedcode from r1) as e natural join r1 ;")

query.append("with r1 as (select licenseno,sum(qty) as totalqty from items natural join storage\_area natural join supply\_record where section\_name='dairy' and date>='2019-05-01' and date <='2019-10-01' group by licenseno) select \* from r1 natural join (select max(totalqty) as totalqty from r1) as e natural join supplier")

query.append("select \* from members natural join (with r1 as (select id,sum(qty) totalqty from members join (select \* from bill natural join bill\_details ) as w on w.customer\_id=members.id where bill\_date>='2019-10-01' and bill\_date<='2019-10-30' group by id ) select id,totalqty from r1 natural join (select max(totalqty) as totalqty from r1) as w) as q")

query.append("with r1 as(select mgrssn,avg(percentage) as totalpresent from employee as e natural join department as d join (select ssn,(((totalpres\*100.00)/totalatt) as percentage from (select ssn,count(is\_present) as totalpres from attendance natural join employee where is\_present=true and date>='2019-10-01' and date<='2019-10-31' group by ssn) as r natural join (select ssn,count(is\_present) as totalatt from attendance natural join employee where date>='2019-10-01' and date<='2019-10-31' group by ssn) as s) as w on w.ssn=e.ssn group by d.mgrssn) select mgrssn,totalpresent,name from r1 natural join (select max(totalpresent) as totalpresent from r1) as v join employee on ssn=mgrssn")

query.append("with r1 as(select age\_group,count(age\_group) as popular from (select itemcode,age\_group from bill\_details natural join (select itemcode,age\_group from items natural join clothes\_description) as i)as fi group by age\_group) select age\_group,popular from (select max(popular) as popular from r1)as f natural join r1;")

query.append("with r1 as (select licenseno, itemcode, cost from supply\_record natural join (select itemcode,min(cost\_price) as cost from supply\_record where date>='2019-10-01' and date<='2019-10-31' group by (itemcode)) as e) select product.productname,supplier.\*,product.brandname,cost from supplier natural join ((select productname,licenseno,cost from r1 natural join personal\_care\_description) union (select productname,licenseno,cost productname from r1 natural join clothes\_description) union (select productname,licenseno,cost productname from r1 natural join packed\_food\_description)) as r2 natural join product;")

try:

```
connection = psycopg2.connect(user="201701060",
                                password="!arunaAmit@777!",
                                host="10.100.71.21",
                                port="5432",
                                database="201701060")

cursor = connection.cursor()

print('connected to database')

postgreSQL_select_Query = "set search_path to supermarket;\n"
```

```
cursor.execute(postgreSQL_select_Query)
```

```
x=1
```

```
while(x is not 0):
```

```
    x=int(input("select query to be executed:\n0.exit\n1.Get item-wise profit of a supermarket
calculated based on the sellout period of each item from date 01-10-2019\n2.Get total profit
generated by supermarket calculated on all transactions made between 10/10/2019 and
10/11/2019\n3.Get the most sold item in the store in the month of october 2019\n4.The details of
the employee who resolved the most compliants in the month of october 2019\n5.Get the
brandname which has the most items sold in the month of october 2019\n6.Get the food item which
has received the most complaints/feedback so far\n7.Get the discount code which was most availed
during the month of october 2019\n8.Get the supplier who has supplied the most number of dairy
products\n9.Get the details of the premium member who has purchased the most number of
products by value in the month of october 2019\n10.Get the manager details of the department
with the highest average attendance percentage in the month of october 2019\n11.Get the age
group for which the supermarket has sold the most clothes\n12.Get a list of all the items and the
suppliers who sold those items at the cheapest rate in the month of october 2019\n"))
```

```
    if(x<=12 and x>=0):
```

```
        if(x==0):
```

```
            print('closing...')
```

```
        else:
```

```
            cursor.execute(query[x-1])
```

```
            print("Selecting rows from query table")
```

```
            queri = cursor.fetchall()
```

```
            column_names=[row[0] for row in cursor.description]
```

```
            print(tabulate(queri, headers=column_names))
```

```
except (Exception, psycopg2.Error) as error :
```

```
    print ("Error while fetching data from PostgreSQL", error)
```

```
finally:
```

```
    #closing database connection.
```

```
    if(connection):
```

```
        cursor.close()
```

```
        connection.close()
```

```
    print("PostgreSQL connection is closed")
```