Filatova Veronika

Database Project
Report

Task 1. Access settings.

In order to associate managers with certain countries they are responsible for, the following queries were used:

```
insert into country_managers (username, country)
values('sophie', 'US');
insert into country_managers (username, country)
values('sophie', 'CA');
insert into country_managers (username, country)
values('kirill', 'FR');
insert into country_managers (username, country)
values('kirill', 'GB');
insert into country_managers (username, country)
values('kirill', 'DE');
insert into country_managers (username, country)
values('kirill', 'AU');
```

Task №2. product2 & country2 materialized views.

Two materialized views (prouct2 and country2) were added:

```
create materialized view product2 as
select
pc.productcategoryid as pcid,
p.productid as productid,
pc."name"  as pcname,
p."name" as pname
from
product as p join productsubcategory as psc
on p.productsubcategoryid = psc.productsubcategoryid
join productcategory as pc
on psc.productcategoryid = pc.productcategoryid

grant select on product2 to planadmin;
grant select on product2 to planmanager;


create materialized view country2 as
select distinct a.countryregioncode
from
address as a join customeraddress as ca
on a.addressid  = ca.addressid
where ca.addresstype = 'Main Office'

grant select on country2 to planadmin;
grant select on country2 to planmanager;
```

Product2 combine data of product and its category. Country2 is filled with unique codes of the countries where shops are located. Managers and administrators are allowed to read from these views.

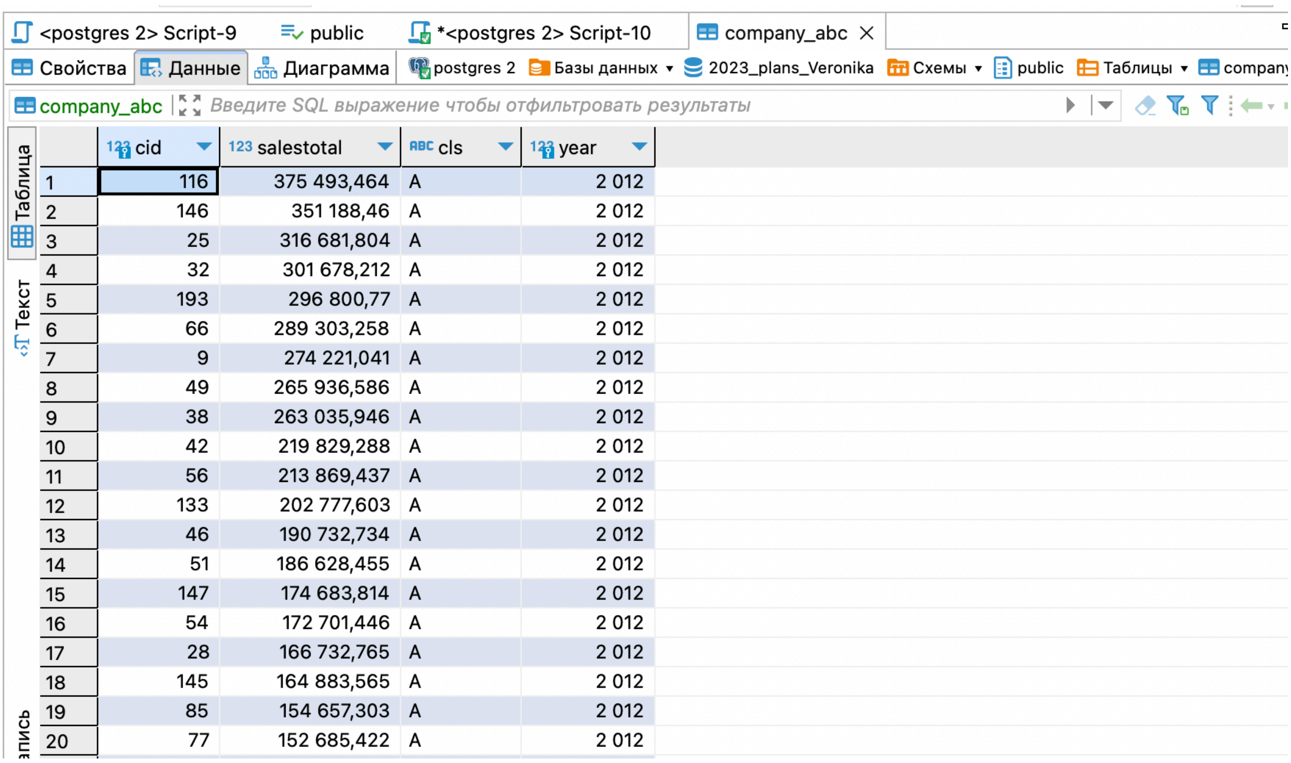
Task №3. Loading data into the company table.

A query to load the country table:

```
insert into company (cname, countrycode, city)
select
c.companyname as cname,
a.countryregioncode as countrycode,
a.city as city
from
customer as c join customeraddress as ca on c.customerid = ca.customerid
join address as a on ca.addressid = a.addressid
where ca.addresstype = 'Main Office’
```

Task №4. Company classification.

The first 20 records from company_abc:



	cid	salestotal	cls	year
1	116	375 493,464	A	2 012
2	146	351 188,46	A	2 012
3	25	316 681,804	A	2 012
4	32	301 678,212	A	2 012
5	193	296 800,77	A	2 012
6	66	289 303,258	A	2 012
7	9	274 221,041	A	2 012
8	49	265 936,586	A	2 012
9	38	263 035,946	A	2 012
10	42	219 829,288	A	2 012
11	56	213 869,437	A	2 012
12	133	202 777,603	A	2 012
13	46	190 732,734	A	2 012
14	51	186 628,455	A	2 012
15	147	174 683,814	A	2 012
16	54	172 701,446	A	2 012
17	28	166 732,765	A	2 012
18	145	164 883,565	A	2 012
19	85	154 657,303	A	2 012
20	77	152 685,422	A	2 012

A designed query for company classification and loading company_abc:

```
insert into company_abc (cid, salestotal, cls, year)
select res.cid,
res.salestotal,
case
      when res.srt <= 0.8*(max(res.srt) over (partition by res.year))
      then 'A'
      when res.srt > 0.8*(max(res.srt) over (partition by res.year)) and res.srt
<= 0.95*(max(res.srt) over (partition by res.year))
      then 'B'
      else 'C'
end as cls,
res.year
from (select *,
coalesce(sum(data.salestotal) over (partition by data.year order by data.year
                rows between unbounded preceding and current row), 0) as srt
from
(select
comp.id as cid,
sum(s.subtotal) as salestotal,
extract (year from s. orderdate) as year
from
salesorderheader as s join customer as c
on s.customerid = c.customerid
join company as comp on c.companyname = comp.cname
where extract (year from s. orderdate) in (2012, 2013)
group by comp.id, year
order by year, salestotal desc) as data) as res
```

Task №5. Finding quarterly sales amount by company and product category.

Query for calculating quarterly sales amount before taxes in 2012 and 2013 for each company and product category:

```
insert into company_sales (cid, salesamt, year, quarter_yr, qr, categoryid,
ccls)
select distinct
c.id as cid,
sum (s2.linetotal) over (partition by c.id, extract (year from s. orderdate),
extract (quarter from s. orderdate), p2.pcid) as salesamt,
extract (year from s. orderdate) as year,
extract (quarter from s. orderdate) as quarter_yr,
to_char(s.orderdate, 'YYYY.Q') as qr,
p2.pcid as categoryid,
ca.cls  as ccls
from
company as c join customer as cu on c.cname = cu.companyname
join salesorderheader as s on s.customerid = cu.customerid
join salesorderdetail as s2 on s2.salesorderid = s.salesorderid
join product2 as p2 on p2.productid = s2.productid
join company_abc ca on ca.cid = c.id and ca.year = extract (year from s.
orderdate)
where year in (2012, 2013)
order by cid, year, quarter_yr, categoryid
```

Task №6. Initial data preparation.

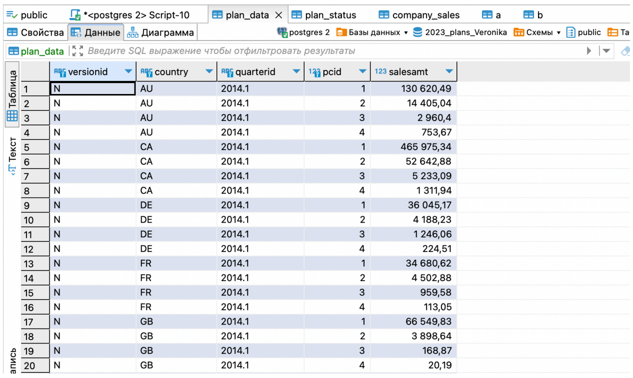
Start planning function:

```
import psycopg2

def start_planning (year, quarter, user, pwd):
    con = psycopg2.connect(database="2023_plans_Veronika", user=user, host='localhost', password = pwd)
    cur = con.cursor()
    qr =  str(year)+'.'+str(quarter)
    query_1 = """delete from plan_data
            where quarterid = %s"""
    val_1 = [qr]
    query_2 = """delete from plan_status
            where quarterid = %s"""
     val_2 = [qr]
    query_3 = """insert into plan_status(quarterid, country, status,
            modifieddatetime, author)
            select distinct
            %s as quarterid,
            c.countrycode as country,
            'R' as status,
            CURRENT_TIMESTAMP as modifieddatetime,
            %s as author
            from company as c"""
    val_3 = [qr, user]
    query_4 = """insert into plan_data
            select distinct
            'N' as versionid,
            c2.countryregioncode as country,
            %s as quarterid,
            cs.categoryid as pcid,
            (sum(cs.salesamt) over (partition by cs.categoryid, c2.countryregioncode))/2 as salesamt
            from country2 as c2 join company as c on c2.countryregioncode = c.countrycode
            join company_sales cs on cs.cid = c.id
            where cs.ccls in ('A', 'B') and cs.year in (%s-1, %s-2) and cs.quarter_yr = %s
            order by country, quarterid, pcid"""
    val_4 = [qr, year, year, quarter]
    query_5 = """insert into plan_data
             SELECT
             'P' as versionid,
             country,quarterid, pcid, salesamt
             FROM plan_data
             where versionid = 'N'"""

    cur.execute(query_1, val_1)
    con.commit()

    cur.execute(query_2, val_2)
    con.commit()

    cur.execute(query_3, val_3)
    con.commit()

    cur.execute(query_4, val_4)
    con.commit()

    cur.execute(query_5)
    con.commit()
```

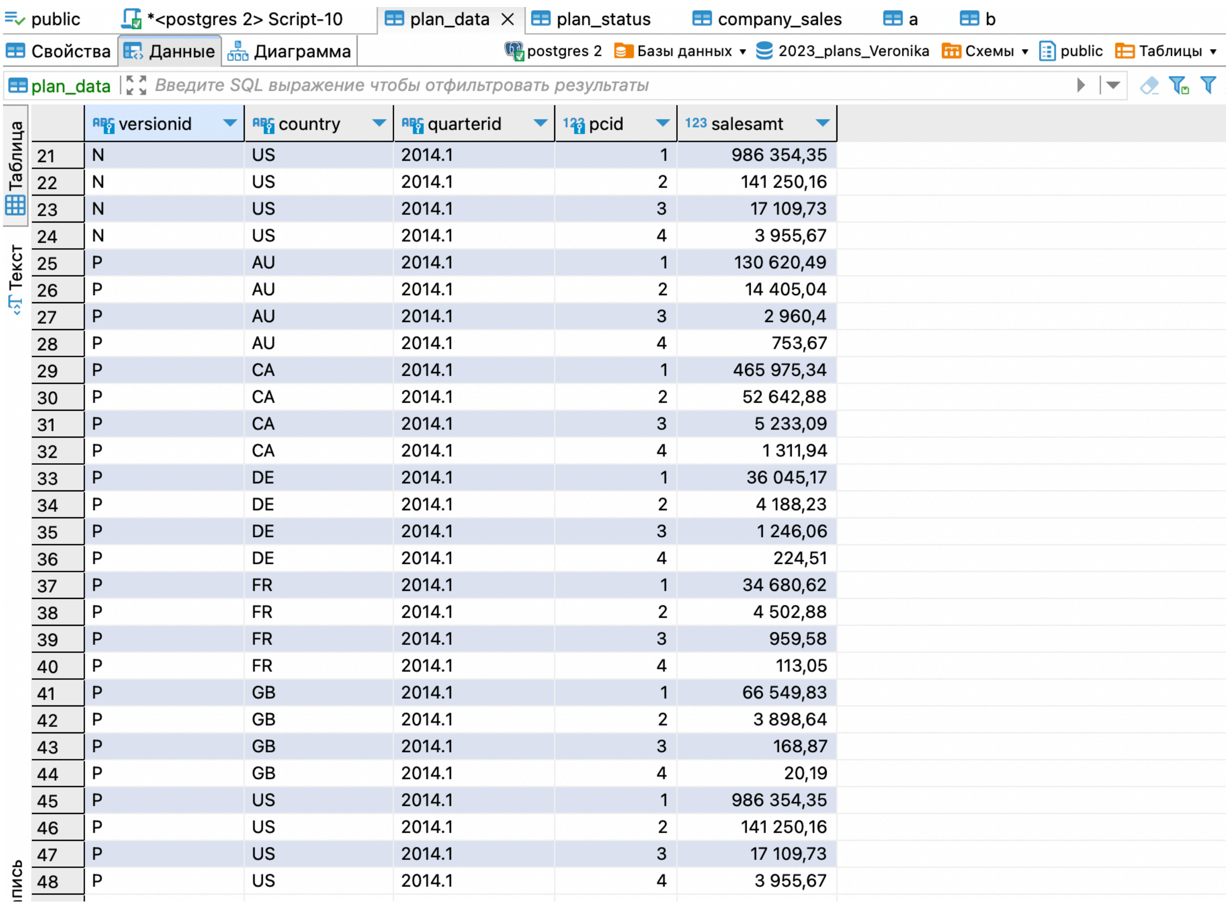
Function call to populate the plan_data and plan_status tables:

start_planning (2014, 1, 'ivan', ‘123')

Plan_data content:

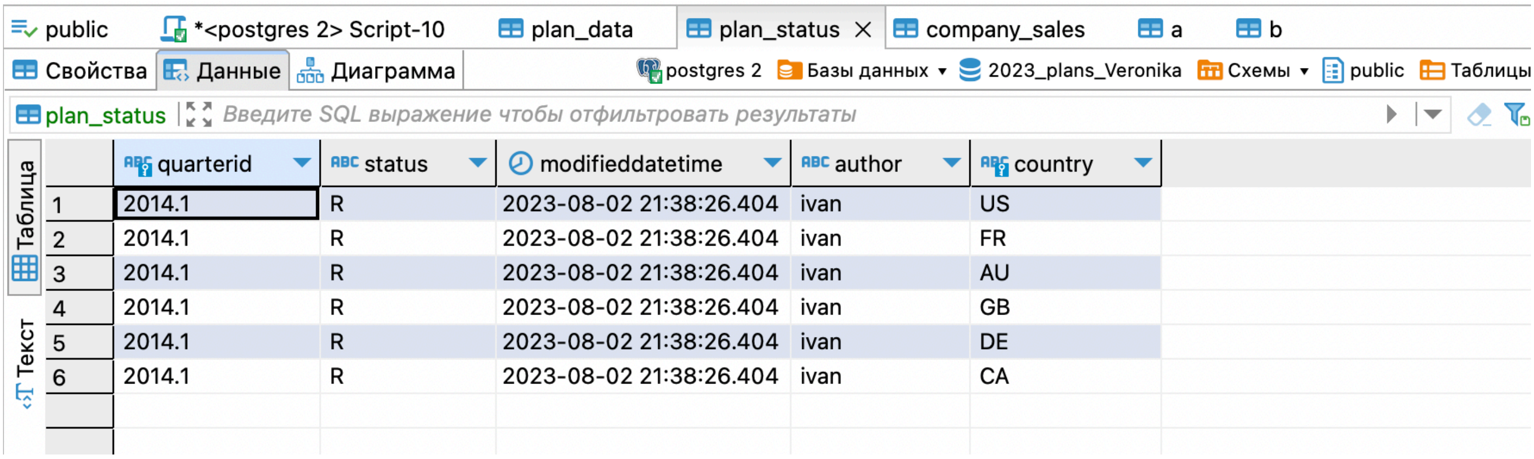


	versionid	country	quarterid	pcid	salesamt
1	N	AU	2014.1	1	130 620,49
2	N	AU	2014.1	2	14 405,04
3	N	AU	2014.1	3	2 960,4
4	N	AU	2014.1	4	753,67
5	N	CA	2014.1	1	465 975,34
6	N	CA	2014.1	2	52 642,88
7	N	CA	2014.1	3	5 233,09
8	N	CA	2014.1	4	1 311,94
9	N	DE	2014.1	1	36 045,17
10	N	DE	2014.1	2	4 188,23
11	N	DE	2014.1	3	1 246,06
12	N	DE	2014.1	4	224,51
13	N	FR	2014.1	1	34 680,62
14	N	FR	2014.1	2	4 502,88
15	N	FR	2014.1	3	959,58
16	N	FR	2014.1	4	113,05
17	N	GB	2014.1	1	66 549,83
18	N	GB	2014.1	2	3 898,64
19	N	GB	2014.1	3	168,87
20	N	GB	2014.1	4	20,19



	versionid	country	quarterid	pcid	salesamt
21	N	US	2014.1	1	986 354,35
22	N	US	2014.1	2	141 250,16
23	N	US	2014.1	3	17 109,73
24	N	US	2014.1	4	3 955,67
25	P	AU	2014.1	1	130 620,49
26	P	AU	2014.1	2	14 405,04
27	P	AU	2014.1	3	2 960,4
28	P	AU	2014.1	4	753,67
29	P	CA	2014.1	1	465 975,34
30	P	CA	2014.1	2	52 642,88
31	P	CA	2014.1	3	5 233,09
32	P	CA	2014.1	4	1 311,94
33	P	DE	2014.1	1	36 045,17
34	P	DE	2014.1	2	4 188,23
35	P	DE	2014.1	3	1 246,06
36	P	DE	2014.1	4	224,51
37	P	FR	2014.1	1	34 680,62
38	P	FR	2014.1	2	4 502,88
39	P	FR	2014.1	3	959,58
40	P	FR	2014.1	4	113,05
41	P	GB	2014.1	1	66 549,83
42	P	GB	2014.1	2	3 898,64
43	P	GB	2014.1	3	168,87
44	P	GB	2014.1	4	20,19
45	P	US	2014.1	1	986 354,35
46	P	US	2014.1	2	141 250,16
47	P	US	2014.1	3	17 109,73
48	P	US	2014.1	4	3 955,67

Plan_status content:



	quarterid	status	modifieddatetime	author	country
1	2014.1	R	2023-08-02 21:38:26.404	ivan	US
2	2014.1	R	2023-08-02 21:38:26.404	ivan	FR
3	2014.1	R	2023-08-02 21:38:26.404	ivan	AU
4	2014.1	R	2023-08-02 21:38:26.404	ivan	GB
5	2014.1	R	2023-08-02 21:38:26.404	ivan	DE
6	2014.1	R	2023-08-02 21:38:26.404	ivan	CA

Task №7. Changing plan data.

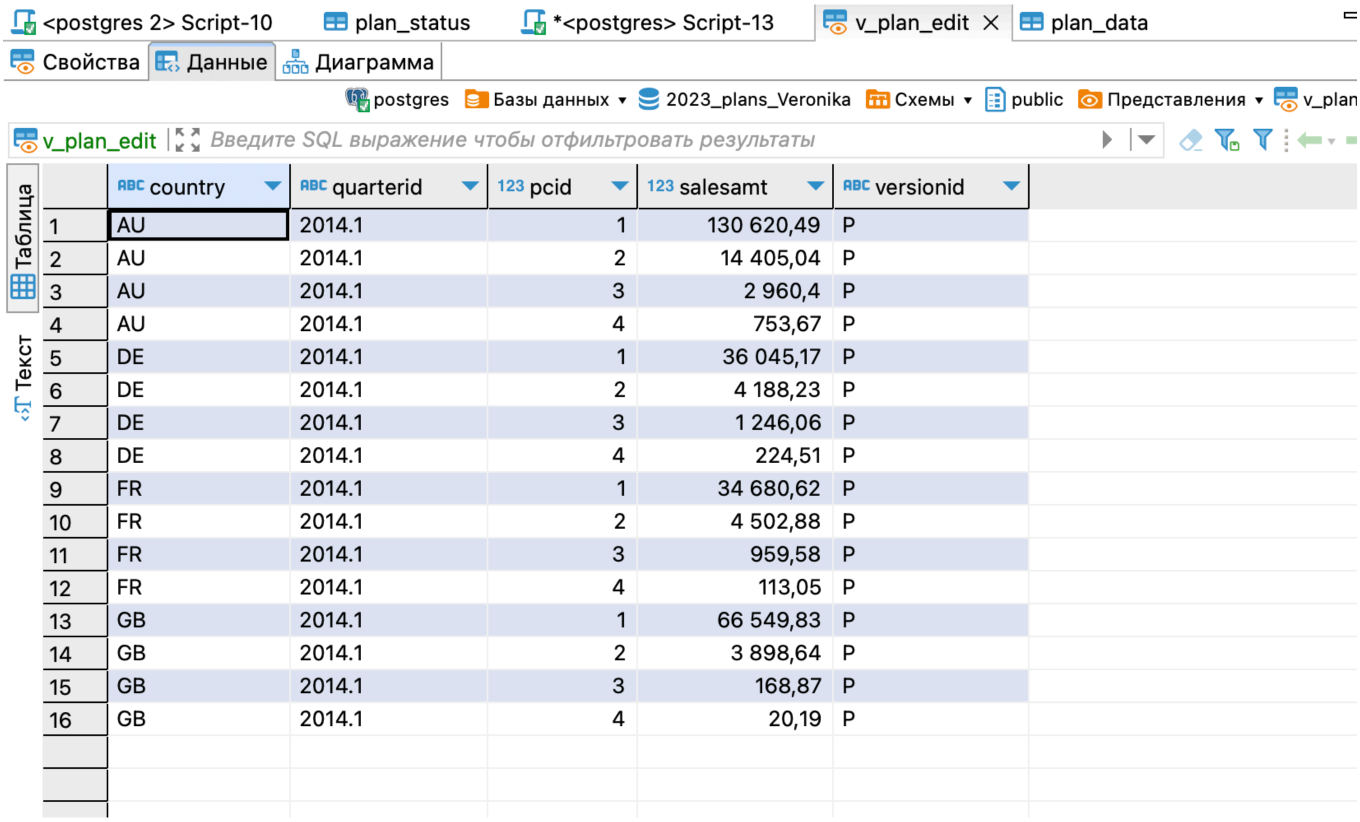
Code for the set_lock function:

```
def set_lock (year, quarter, user, pwd):
    con = psycopg2.connect(database="2023_plans_Veronika", user=user, host='localhost', password = pwd)
    cur = con.cursor()
    qr =  str(year)+'.'+str(quarter)
    query_1 = """UPDATE
    plan_status
    SET
    status = 'L',
    modifieddatetime = CURRENT_TIMESTAMP,
    author = CURRENT_USER
    where country in (select country
    from country_managers cm
    where username = CURRENT_USER
    and quarterid = %s)"""
    val_1 = [qr]
    cur.execute(query_1, val_1)
    con.commit()
```

Code for the remove_lock function:

```
def remove_lock (year, quarter, user, pwd):
    con = psycopg2.connect(database="2023_plans_Veronika", user=user, host='localhost', password = pwd)
    cur = con.cursor()
    qr =  str(year)+'.'+str(quarter)
    query_1 = """UPDATE
    plan_status
    SET
    status = 'R',
    modifieddatetime = CURRENT_TIMESTAMP,
    author = CURRENT_USER
    where country in (select country
    from country_managers cm
    where username = CURRENT_USER
    and quarterid = %s)"""
    val_1 = [qr, user]
    cur.execute(query_1, val_1)
    con.commit()
```

A screenshot of v_plan_edit contents when logged in as kirill:



	country	quarterid	pcid	salesamt	versionid
1	AU	2014.1	1	130 620,49	P
2	AU	2014.1	2	14 405,04	P
3	AU	2014.1	3	2 960,4	P
4	AU	2014.1	4	753,67	P
5	DE	2014.1	1	36 045,17	P
6	DE	2014.1	2	4 188,23	P
7	DE	2014.1	3	1 246,06	P
8	DE	2014.1	4	224,51	P
9	FR	2014.1	1	34 680,62	P
10	FR	2014.1	2	4 502,88	P
11	FR	2014.1	3	959,58	P
12	FR	2014.1	4	113,05	P
13	GB	2014.1	1	66 549,83	P
14	GB	2014.1	2	3 898,64	P
15	GB	2014.1	3	168,87	P
16	GB	2014.1	4	20,19	P

Task №8. Plan data approval.

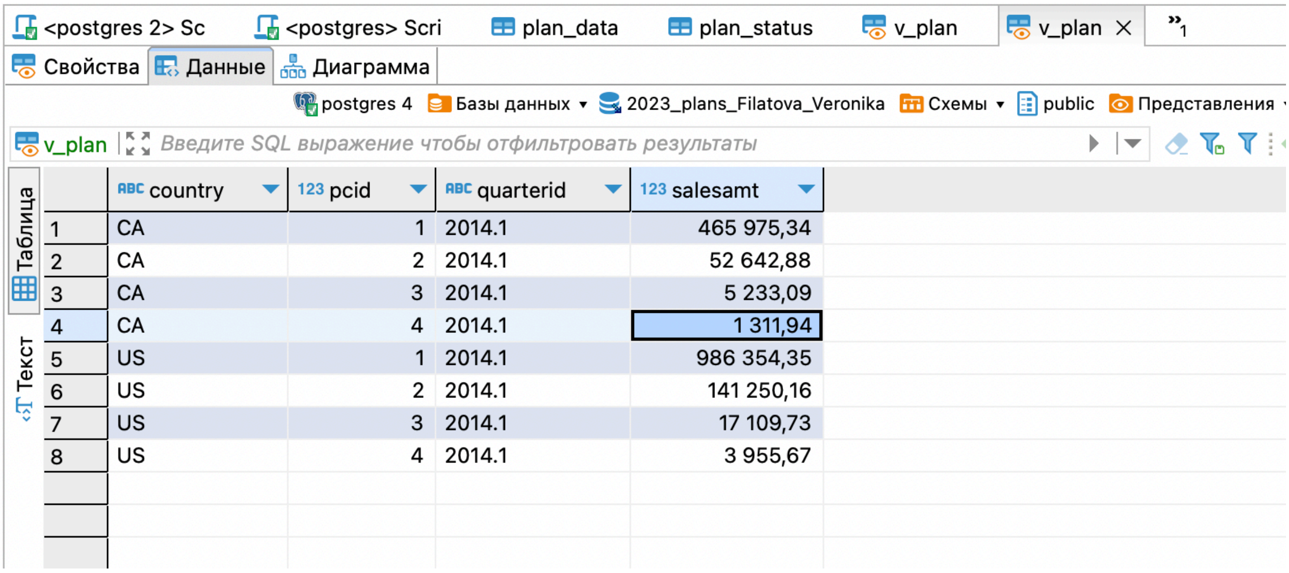
Code of the accept_plan function:

```
def accept_plan(year, quarter, user, pwd):
    con = psycopg2.connect(database="2023_plans_Filatova_Veronika", user=user, host='localhost', password = pwd)
    cur = con.cursor()
    qr =  str(year)+'.'+str(quarter)

    query_1 = """delete from plan_data
                where quarterid = %s and versionid = 'A' and
                country in (select country
                from country_managers cm
                where username = CURRENT_USER)"""
    val_1 = [qr]
    query_2 = """insert into plan_data
    select 'A' as versionid,
    country, quarterid, pcid, salesamt
    from plan_data
    where quarterid = %s
    and country in (select cm.country
    from country_managers cm join plan_status ps
    on cm.username = ps.author
    where cm.username = CURRENT_USER and ps.status = 'R' )
    and versionid = 'P'"""
    val_2 = [qr]
    query_3 = """update plan_status
    set
    status = 'A',
    modifieddatetime = CURRENT_TIMESTAMP,
    author = CURRENT_USER
    where country in (select ps.country from plan_data as pd
    join plan_status ps on pd.country = ps.country
    join country_managers cm on cm.country = ps.country
    where pd.versionid = 'P' and pd.quarterid = %s
    and cm.username = CURRENT_USER)"""
    val_3 = [qr]

    cur.execute(query_1, val_1)
    con.commit()
    cur.execute(query_2, val_2)
    con.commit()
    cur.execute(query_3, val_3)
    con.commit()
```

Function calls as kirill and Sophie:

```
accept_plan (2014, 1, 'kirill', ‘789')
accept_plan  (2014, 1, 'sophie', '456')
```

V_plan view after logging in as Sophie:



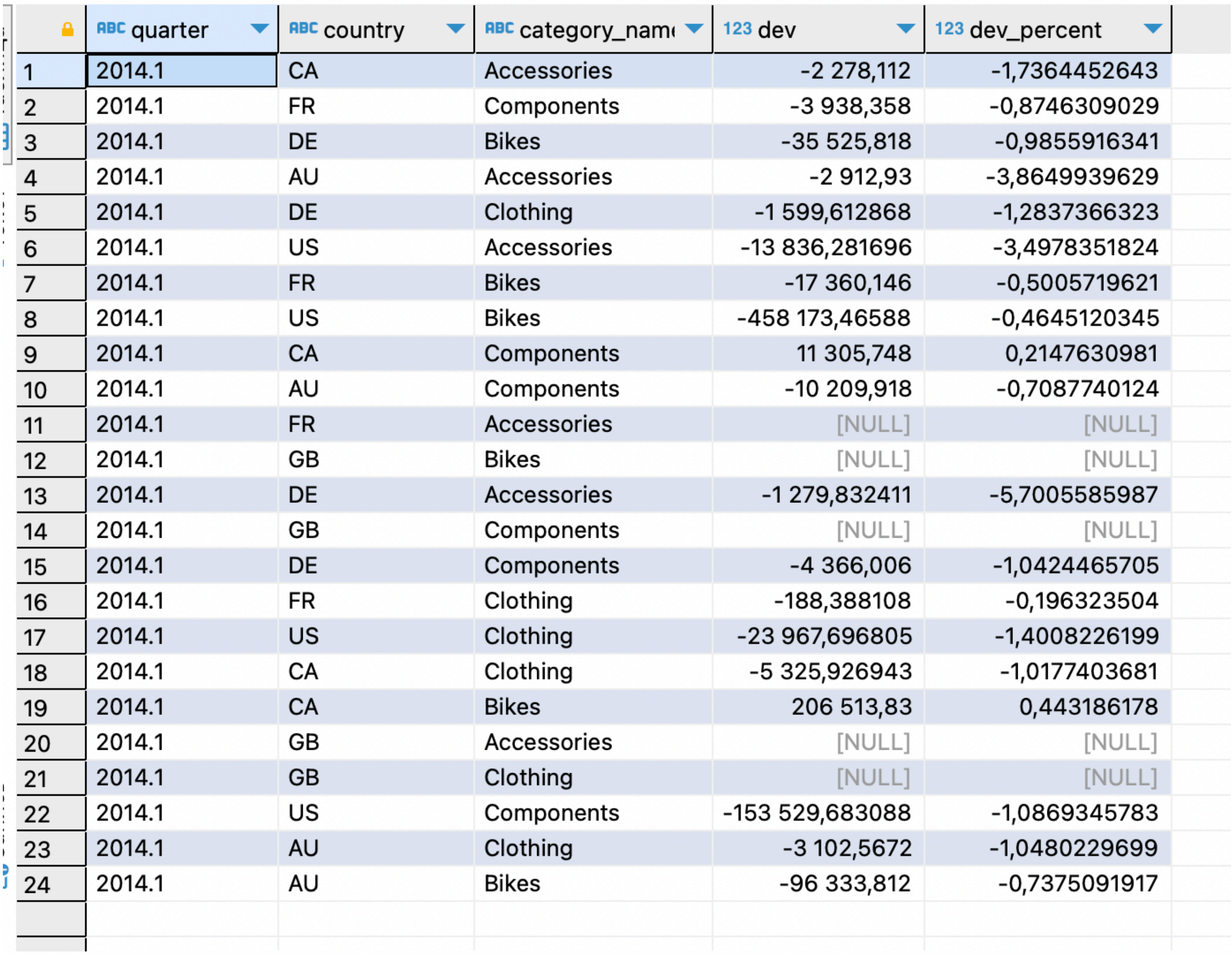
	country	pcid	quarterid	salesamt
1	CA	1	2014.1	465 975,34
2	CA	2	2014.1	52 642,88
3	CA	3	2014.1	5 233,09
4	CA	4	2014.1	1 311,94
5	US	1	2014.1	986 354,35
6	US	2	2014.1	141 250,16
7	US	3	2014.1	17 109,73
8	US	4	2014.1	3 955,67

Task №9. Data preparation for plan-fact analysis in Q1 2014.

The actual data was calculated using salesorderheader and ordersalesdetail tables (2 approach). SQL code of the new materialized view:

```
create materialized view mv_plan_fact_2014_q1 as
select
plan.quarter,
plan.country,
plan.category_name,
plan.plan-fact.fact as dev,
(plan.plan-fact.fact)/plan.plan as dev_percent
from
(select distinct
pd.quarterid as quarter,
pd.country as country ,
p2.pcname as category_name,
pd.salesamt as plan
from plan_data as pd
join product2 as p2
on pd.pcid = p2.pcid
where pd.versionid = 'A' and pd.quarterid = '2014.1') as plan
left join
(select distinct
to_char(sh.orderdate, 'YYYY.Q') as quarter,
com.countrycode as country,
p2.pcname as category_name,
sum(st.linetotal) over (partition by to_char(sh.orderdate,
'YYYY.Q'),com.countrycode,p2.pcname) as fact
from salesorderdetail as st
join salesorderheader as sh
on st.salesorderid = sh.salesorderid
join customer c on sh.customerid = c.customerid
join company as com
on c.companyname = com.cname
join product2 as p2 on
st.productid = p2.productid
where to_char(sh.orderdate, 'YYYY.Q') = '2014.1'
and com.id in (select c.id
from company_abc as c_a
join company as c
on c_a.cid = c.id
where c_a.cls in ('A', 'B')
and c_a.year = 2013)) as fact
on fact.quarter = plan.quarter
and fact.country = plan.country
and fact.category_name = plan.category_name
```

Data in mv_plan_fact_2014_ql view:



	quarter	country	category_nam	dev	dev_percent
1	2014.1	CA	Accessories	-2 278,112	-1,7364452643
2	2014.1	FR	Components	-3 938,358	-0,8746309029
3	2014.1	DE	Bikes	-35 525,818	-0,9855916341
4	2014.1	AU	Accessories	-2 912,93	-3,8649939629
5	2014.1	DE	Clothing	-1 599,612868	-1,2837366323
6	2014.1	US	Accessories	-13 836,281696	-3,4978351824
7	2014.1	FR	Bikes	-17 360,146	-0,5005719621
8	2014.1	US	Bikes	-458 173,46588	-0,4645120345
9	2014.1	CA	Components	11 305,748	0,2147630981
10	2014.1	AU	Components	-10 209,918	-0,7087740124
11	2014.1	FR	Accessories	[NULL]	[NULL]
12	2014.1	GB	Bikes	[NULL]	[NULL]
13	2014.1	DE	Accessories	-1 279,832411	-5,7005585987
14	2014.1	GB	Components	[NULL]	[NULL]
15	2014.1	DE	Components	-4 366,006	-1,0424465705
16	2014.1	FR	Clothing	-188,388108	-0,196323504
17	2014.1	US	Clothing	-23 967,696805	-1,4008226199
18	2014.1	CA	Clothing	-5 325,926943	-1,0177403681
19	2014.1	CA	Bikes	206 513,83	0,443186178
20	2014.1	GB	Accessories	[NULL]	[NULL]
21	2014.1	GB	Clothing	[NULL]	[NULL]
22	2014.1	US	Components	-153 529,683088	-1,0869345783
23	2014.1	AU	Clothing	-3 102,5672	-1,0480229699
24	2014.1	AU	Bikes	-96 333,812	-0,7375091917