

SQL Course Final Task by Alexey Nepochatov. HSE MDS. 2022.

Content:

• Before begin. Database preparation.....	2
• Task №1. Access settings.	3
• Task №2. product2 & country 2 materialized views.....	5
• Task №3. Loading data into the company table.	7
• Task №4. Company classification.	8
• Task №5. Finding quarterly sales amount by company, and product category.	9
• Task №6. Initial data preparation.	10
• Task №7. Changing plan data.	12
• Task №8. Plan data approval.	15
• Task №9. Data preparation for plan-fact analysis in Q1 2014.	17



NATIONAL RESEARCH
UNIVERSITY

Before begin. Database preparation.

Database:	2022_plans_alexey_nepochatov.dump + "FT. Tables, views and roles – DDL"
CMD command:	<pre>cd C:\Program Files\PostgreSQL\14\bin pg_restore -h localhost -U postgres -d 2022_plans_alexey_nepochatov -v "C:\Users\Alexey\Desktop\HSE\01 Programming\06 SQL\Week 6\Final Task\Final_Task_Nepochatov\Files_for_task_adventureworks1t.dump"</pre>
<p>The diagram illustrates the database schema with the following tables and their attributes:</p> <ul style="list-style-type: none"> address: addressid (PK), addressline1, addressline2, city, postalcode, spatiallocation, stateprovinceid, countryregioncode, isonlystateprovinceflag, stateprovincename, territoryid, territoryname customeraddress: customerid (FK), addressid (FK), addresstype, modifieddate customer: customerid (PK), namestyle, title, firstname, middleinitial, lastname, suffix, companyname, emailaddress, rowguid, modifieddate, salespersonid, gender, totalchildren, birthdate, datefirstpurchase, persontype salesorderheader: salesorderid (PK), revisionnumber, orderdate, duedate, shipdate, status, onlineorderflag, salesordernumber, purchaseordernumber, accountnumber, customerid (FK), shiptoaddressid, billtoaddressid, shipmethod, creditcardapprovalcode, subtotal, taxamt, freight, totaldue, comment, rowguid, modifieddate salesorderdetail: salesorderid (FK), salesorderdetailid (PK), orderqty, productid (FK), unitprice, unitpricediscount, linetotal, rowguid, modifieddate productsubcategory: productsubcategoryid (PK), productcategoryid (FK), productname product: productid (PK), name, productnumber, color, listprice, size, sizeunitmeasurecode, weightunitmeasurecode, weight, productline, class, style, productsubcategoryid (FK), productmodelid, sellstartdate, sellenddate, discontinueddate productcategory: productcategoryid (PK), name productdescription: productdescriptionid (PK), description, rowguid, modifieddate productphoto: productphotoid (PK), thumbnailphoto, thumbnailphotofilename, largephoto, largephotofilename, modifieddate productproductphoto: productid (FK), productphotoid (FK), modifieddate 	
Figure 1. Tables with business data	

Task №1. Access settings.

```
grant select on all tables in schema public to planadmin;  
grant select, insert, update, delete on plan_data to planadmin;  
grant select, insert, update, delete on plan_status to planadmin;  
grant select, insert, update, delete on country_managers to planadmin;  
revoke select on public.v_plan, public.v_plan_edit from planadmin;
```

```
grant select on all tables in schema public to planmanager;  
grant select, insert, update, delete on plan_data to planmanager;  
grant select, update on plan_status to planmanager;  
grant select on country_managers to planmanager;  
grant select on v_plan to planmanager;  
grant select, update on v_plan_edit to planmanager;
```

```
create user ivan with password 'password';  
create user sophie with password 'password';  
create user kirill with password 'password';
```

```
grant planadmin to ivan;  
grant planmanager to sophie;  
grant planmanager to kirill;
```

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

Name	Value
Updated Rows	6
Query	<pre>grant select on all tables in schema public to planadmin; grant select, insert, update, delete on plan_data to planadmin; grant select, insert, update, delete on plan_status to planadmin; grant select, insert, update, delete on country_managers to planadmin; revoke select on public.v_plan, public.v_plan_edit from planadmin; grant select on all tables in schema public to planmanager; grant select, insert, update, delete on plan_data to planmanager; grant select, update on plan_status to planmanager; grant select on country_managers to planmanager; grant select on v_plan to planmanager; grant select, update on v_plan_edit to planmanager; create user ivan with password 'password'; create user sophie with password 'password'; create user kirill with password 'password'; grant planadmin to ivan; grant planmanager to sophie; grant planmanager to kirill; insert into country_managers (username, country) values ('sophie', 'US'), ('sophie', 'CA'), ('kirill', 'FR'), ('kirill', 'GB'), ('kirill', 'DE'), ('kirill', 'AU');</pre>
Finish time	Mon Aug 15 13:18:31 MSK 2022

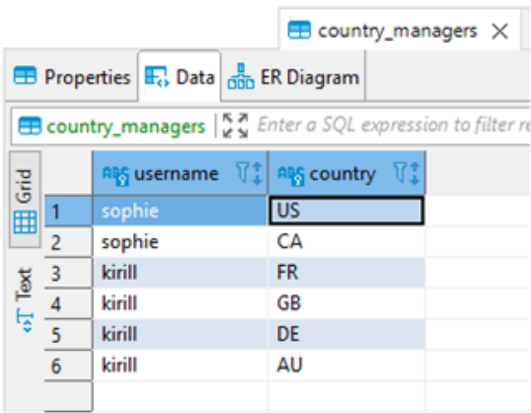


Figure 2. The result of successful completion of query and “country_managers” table.

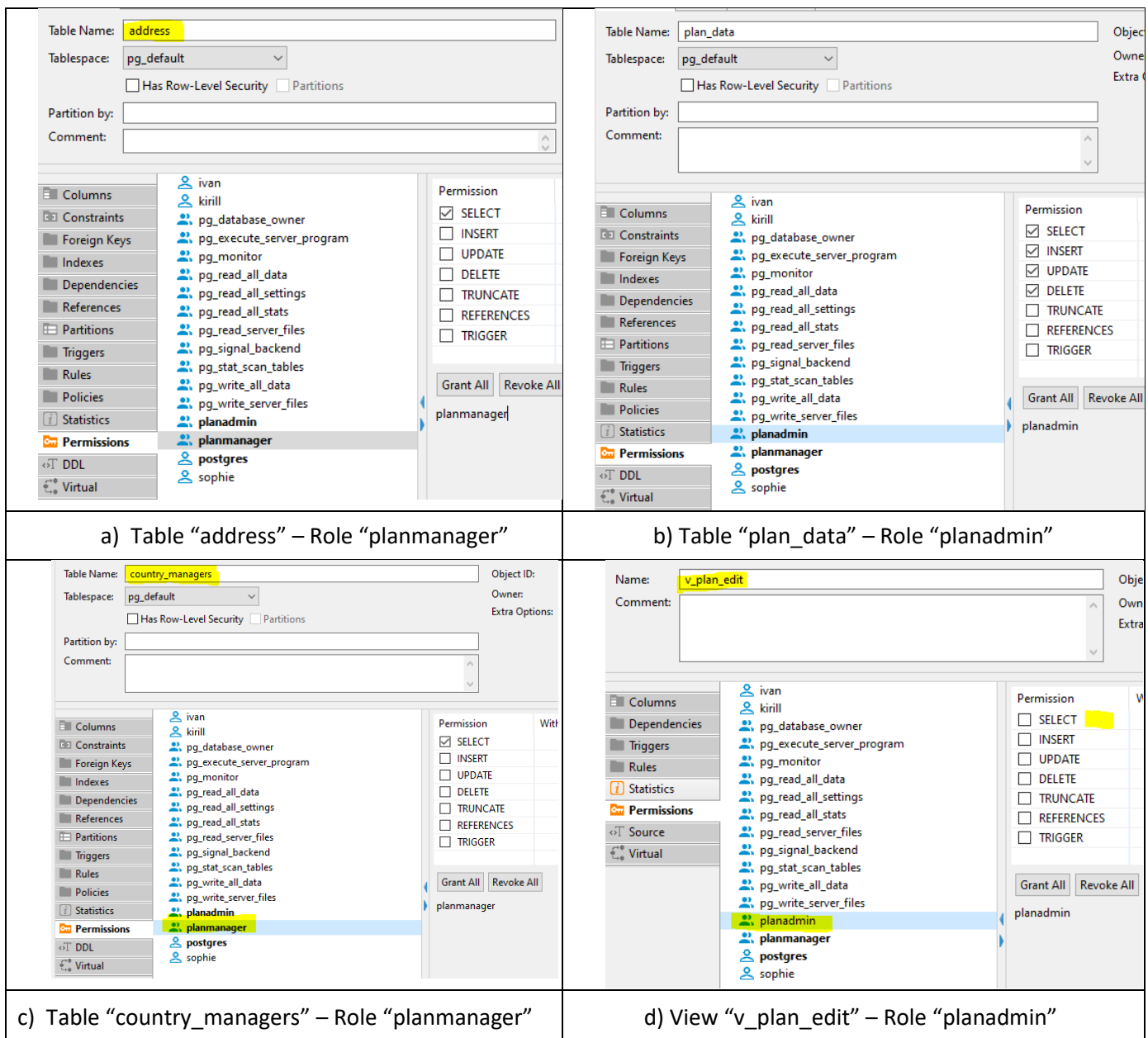
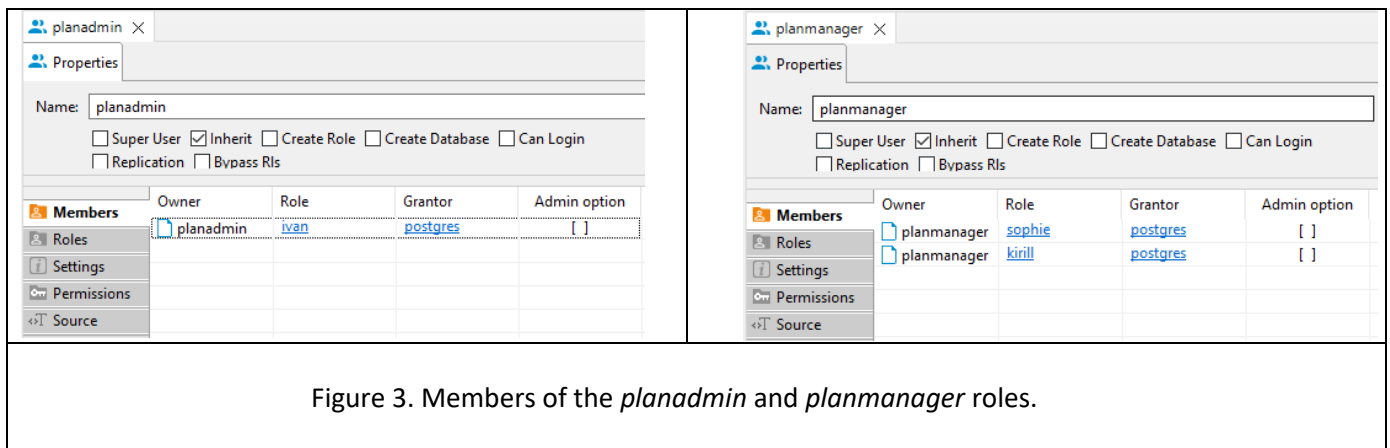


Figure 4. Examples of permissions for different tables and roles.

Task №2. product2 & country 2 materialized views.

The 'product2' view query:

```
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
with no data;
```

refresh materialized view product2;

The 'country2' view query:

```
create materialized view country2 as
select distinct a.countryregioncode as countrycode
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 in ('Main Office')
with no data;
```

refresh materialized view country2; Query to allow managers and administrators to read from these views:

```
grant select on product2 to planadmin;
grant select on product2 to planmanager;
grant select on country2 to planadmin;
grant select on country2 to planmanager;
```

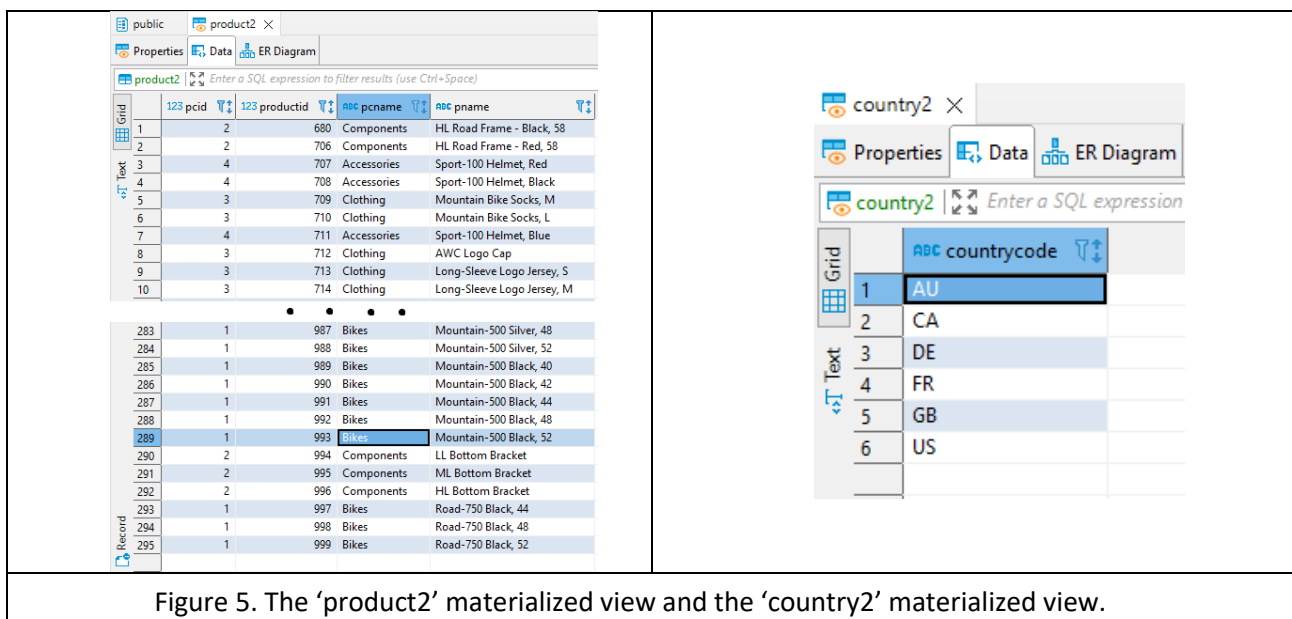


Figure 5. The 'product2' materialized view and the 'country2' materialized view.

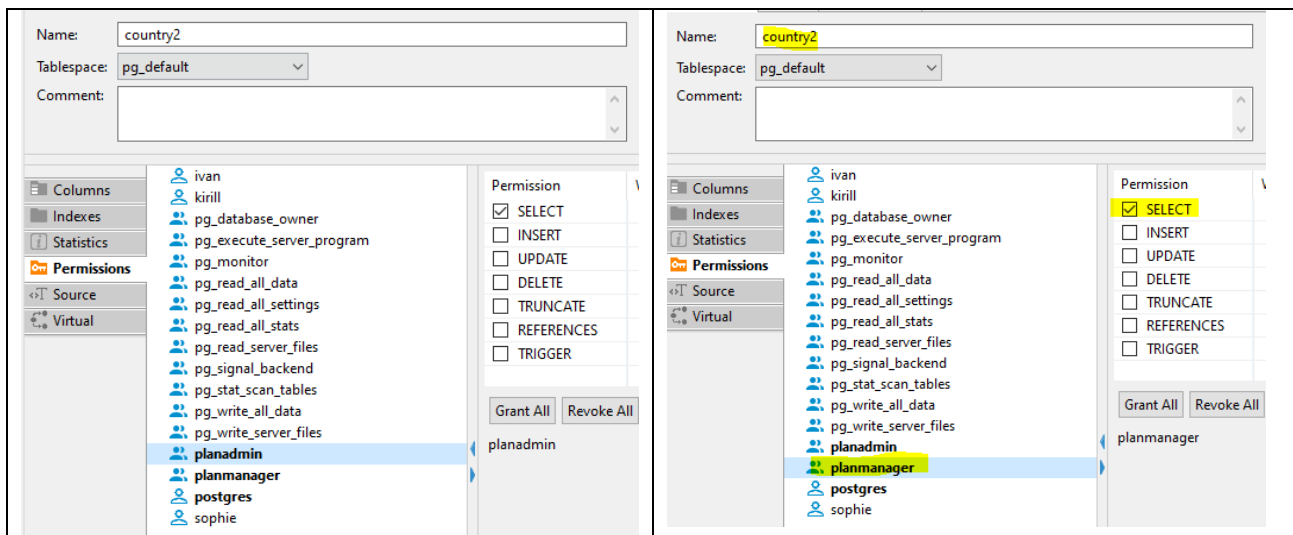


Figure 6. Permissions for “country2” materialized view.

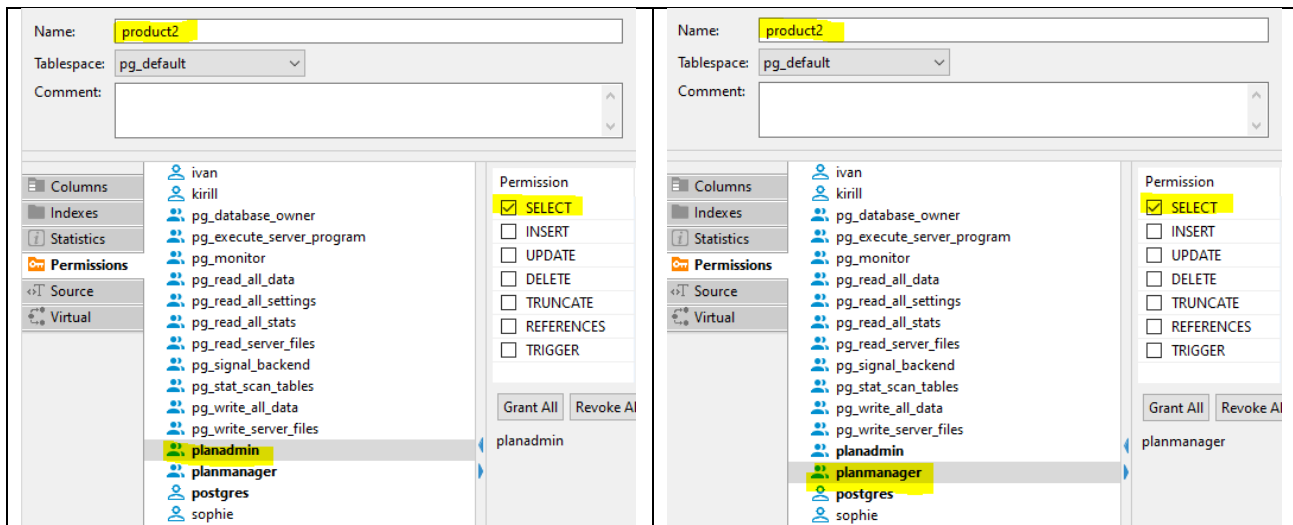


Figure 7. Permissions for “product2” materialized view.

Task №3. Loading data into the company table.

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

company					
Enter a SQL expression to filter results (use Ctrl+Space)					
Grid	id	ABC cname	ABC countrycode	ABC city	
1	1	A Bicycle Association	CA	Richmond	
2	2	A Cycle Shop	US	Southfield	
3	3	Acceptable Sales & Service	US	Culver City	
4	4	Action Bicycle Specialists	US	Torrance	
5	5	Active Cycling	AU	North Sydney	
6	6	Activity Center	US	Central Valley	
7	7	Alpine Ski House	US	Portland	
8	8	Amalgamated Parts Shop	US	Hooksett	
9	9	Area Sheet Metal Supply	US	Everett	
10	10	Basic Sports Equipment	US	El Segundo	
11	11	Beneficial Exercises and Activities	US	Southgate	
12	12	Best Cycle Store	US	Modesto	
13	13	Better Bike Shop	AU	North Sydney	
14	14	Bicycle Accessories and Kits	US	Redford	
15	15	Bike Boutique	AU	Melbourne	
16	16	Bike Part Wholesalers	AU	Hawthorne	
17	17	Bikes Anyone?	US	Dallas	
...					
190	190	Undergraze and Finish Company	US	Stockton	
191	191	Utilitarian Sporting Goods	US	Racine	
192	192	Uttermost Bike Shop	US	Bothell	
193	193	Valley Bicycle Specialists	US	Waterbury	
194	194	Valuable Bike Parts Company	US	Farmington	
195	195	Variety Cycling	US	San Diego	
196	196	Versatile Sporting Goods Company	US	Vacaville	
197	197	Vigorous Exercise Company	CA	Toronto	
198	198	West Wind Distributors	CA	Calgary	
199	199	Western Bike Supplies	US	Biloxi	
200	200	Wingtip Toys	US	Downey	
201	201	Work and Play Association	US	Leesburg	
202	202	World Bike Discount Store	CA	Richmond Hill	
203	203	Worthwhile Activity Store	US	North Sioux City	
204	204	Year-Round Sports	US	Miami	
205	205	Yellow Bicycle Company	DE	Bonn	

Figure 8. Data loaded into “company” table (205 rows).

Task №4. Company classification.

```

insert into company_abc
select s.cid,
       s.salestotal as salestotal,
       case
         when ((sum(s.salestotal) over(partition by s."year" order by s.salestotal desc))
/ (sum(s.salestotal) over(partition by s."year"))) <= 0.8 then 'A'
         when ((sum(s.salestotal) over(partition by s."year" order by s.salestotal desc))
/ (sum(s.salestotal) over(partition by s."year"))) <= 0.95 then 'B'
         else 'C'
       end as cls,
       s.year
from (
  select c.id as cid,
         extract(year from orderdate) as "year",
         sum(so.subtotal) as "salestotal"
  from company as c
       join customer as cs
         on cs.companyname = c.cname
       join salesorderheader as so
         on so.customerid = cs.customerid
       where extract(year from so.orderdate) in (2012, 2013)
  group by c.id,
           extract(year from orderdate)
) as s;

```

	cid	salestotal	cls	year
1	1	838.9178	C	2,012
2	4	164,052.2313	A	2,013
3	5	1,409.382	C	2,012
4	5	396.072	C	2,013
5	6	16,948.4192	C	2,012
6	6	20,553.7781	C	2,013
7	7	10,322.5505	C	2,012
8	7	604.872	C	2,013
9	8	140,039.4025	A	2,013
10	9	1,696.1436	C	2,012
11	9	728.658	C	2,013
12	10	2,732.286	C	2,013
13	10	28,259.2575	B	2,012
14	11	794.2467	C	2,012
15	11	896.154	C	2,013
16	12	208,290.8774	A	2,013
17	13	146,262.05	A	2,013
18	13	124,286.3434	A	2,012
19	14	405.585	C	2,013
20	14	6,181.7913	C	2,012
278	197	252,613.2865	A	2,013
279	197	316,681.8038	A	2,012
280	198	12,052.1894	C	2,013
281	199	13,245.2607	C	2,012
282	199	6,712.194	C	2,013
283	200	1,081.818	C	2,013
284	201	784.2007	C	2,012
285	201	73,870.692	B	2,013
286	202	527.4748	C	2,012
287	202	3,132.672	C	2,013
288	203	102,690.9128	A	2,013
289	203	598.641	C	2,012
290	204	81,132.1446	B	2,012
291	205	41.994	C	2,013
292	205	40,766.1068	B	2,012

Figure 9. Data loaded into 'company_abc' table (292 rows).

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

```

insert into company_sales
select
    c.id as cid,
    sum(sod.linetotal) as salesamt,
    extract('year' from soh.orderdate) as "year",
    extract('quarter' from soh.orderdate) as quarter_yr,
    to_char(soh.orderdate, 'YYYY.Q') as qr,
    p_2.pcid as categoryid,
    c_abc.cls as ccls
from
    salesorderheader as soh
join salesorderdetail as sod
    using (salesorderid)
join customer as cst
    using(customerid)
join company as c
    on cst.companyname = c.cname
join product2 as p_2
    using(productid)
join company_abc as c_abc
    on c_abc.cid = c.id
    where c_abc."year" = extract('year' from soh.orderdate)
group by extract('year' from soh.orderdate),
    extract('quarter' from soh.orderdate),
    to_char(soh.orderdate, 'YYYY.Q'),
    c.id,
    p_2.pcid,
    c_abc.cls
order by cid;

```

-- Company key
-- Total amount sold
-- Year
-- Quarter of year
-- Quarter
-- Product category's key
-- Company's class code

cid	salesamt	year	quarter_yr	qr	categoryid	ccls
1	838.92	2,012	1	2012.1	1	C
2	59,060.72	2,013	3	2013.3	1	A
3	28,023.86	2,013	3	2013.3	2	A
4	3,786.6	2,013	3	2013.3	3	A
5	1,406.88	2,013	3	2013.3	4	A
6	55,561.95	2,013	4	2013.4	1	A
7	12,991.91	2,013	4	2013.4	2	A
8	2,084.31	2,013	4	2013.4	3	A
9	1,136.02	2,013	4	2013.4	4	A
10	1,409.38	2,012	4	2012.4	1	C
11	396.07	2,013	1	2013.1	2	C
12	4,557	2,012	2	2012.2	1	C
13	1,951.6	2,012	2	2012.2	2	C
14	389.92	2,012	2	2012.2	3	C
15	105.98	2,012	2	2012.2	4	C
16	5,637.53	2,012	3	2012.3	1	C
17	67.54	2,012	3	2012.3	2	C
18	480.5	2,012	3	2012.3	3	C
19	3,758.35	2,012	4	2012.4	1	C
20	3,288.56	2,013	1	2013.1	1	C
21	919.06	2,013	1	2013.1	2	C
22	361.08	2,013	1	2013.1	3	C
23	9,641.14	2,013	3	2013.3	1	C
2058	2,689.18	2,013	4	2013.4	1	C
2059	180.13	2,012	2	2012.2	2	C
2060	209.26	2,012	3	2012.3	2	C
2061	209.26	2,012	4	2012.4	2	C
2062	38,461.25	2,013	2	2013.2	1	A
2063	7,581,411	2,013	2	2013.2	2	A
2064	377.95	2,013	2	2013.2	3	A
2065	25,462	2,013	3	2013.3	1	A
2066	5,521.23	2,013	3	2013.3	2	A
2067	335.95	2,013	3	2013.3	3	A
2068	22,223.51	2,013	4	2013.4	1	A
2069	1,844.5	2,013	4	2013.4	2	A
2070	883.1	2,013	4	2013.4	3	A
2071	71,249.79	2,012	1	2012.1	1	B
2072	7,368.72	2,012	1	2012.1	2	B
2073	34.2	2,012	1	2012.1	3	B
2074	20.52	2,012	2	2012.2	2	B
2075	2,458.92	2,012	3	2012.3	1	B
2076	40,664.88	2,012	1	2012.1	1	B
2077	101.23	2,012	1	2012.1	3	B
2078	41.99	2,013	4	2013.4	3	C

Figure 10. Data loaded into 'company_sales' table (2078 rows).

Task №6. Initial data preparation.

```
import psycopg2

def start_planning(year, quarter, user, pwd):
    con = psycopg2.connect(database='2022_plans_alexey_nepochatov',
                           user=user,
                           password=pwd,
                           host='localhost')

    cur = con.cursor()
    year_quarter = str(year) + '.' + str(quarter)

    try:
        # Delete plan data from the plan_data table related to the target year and quarter.
        cur.execute(f"""delete from plan_data as pd where pd.quarterid in ({year_quarter}::text)""")
        # In the plan_status table delete records related to the target quarter
        cur.execute(f"""delete from plan_status as ps where ps.quarterid in ({year_quarter}::text)""")

        # Create planning status records (plan_status table) for the selected quarter.
        cur.execute(f"""
insert into plan_status(quarterid, status, modifieddatetime, author, country)
select
    {year_quarter}::text as quarterid, -- Key of planning quarter
    'R' as status, -- Planning data slice status
    now() as modifieddatetime, -- Time when the record was changed-created
    user as author, -- User that changed the record
    countrycode as country -- Country of a shop
from country2;""")
        # Generate version N of planning data in the plan_data table
        cur.execute(f"""
insert into plan_data
select 'N' as versionid, -- Version of plan
    country_product.country as country, -- Country of a shop
    {year_quarter}::text as quarterid, -- Key of planning quarter
    country_product.pcid as pcid, -- Product category's key
    coalesce(avg(s.salesamt),0) as salesamt -- Sales amount before taxes
from (
    select c2.countrycode as country,
        p.pcid
    from country2 as c2
    cross join product2 as p
        where p.pcid is not null
    ) as country_product
join (
    select c.countrycode as country,
        cs.qr as quarterid,
        cs.categoryid as pcid,
        sum(cs.salesamt) as salesamt
    from company as c
    join company_sales as cs
        on cs.cid=c.id
        where cs.ccls in('A', 'B')
        and cs."year" in(({year}-2, {year}-1)
        and cs.quarter_yr in(({quarter}))
    group by c.countrycode ,
        cs.qr,
        cs.categoryid
    ) as s
        using(country)
        where s.pcid=country_product.pcid
    group by country_product.country,
        country_product.pcid
    order by 2, 4
""")

        cur.execute(f"""
insert into plan_data (versionid, country, quarterid, pcid, salesamt)
select 'P' as versionid,
    pd.country as country,
    pd.quarterid as quarterid,
    pd.pcid as pcid,
    pd.salesamt as salesamt
from plan_data as pd
where pd.quarterid in(({year_quarter}::text)
    and versionid in('N');
""")
```

```

finally:
    con.commit()
    con.close()

if __name__ == '__main__':
    start_planning(2014, 1, 'ivan', 'password')

```

versionid	country	quarterid	pcid	salesamt
1	AU	2014.1	1	130,620.49
2	AU	2014.1	2	14,405.04
3	AU	2014.1	3	2,960.4
4	AU	2014.1	4	753.67
5	CA	2014.1	1	465,975.34
6	CA	2014.1	2	52,642.88
7	CA	2014.1	3	5,233.09
8	CA	2014.1	4	1,311.94
9	DE	2014.1	1	36,045.17
10	DE	2014.1	2	8,376.45
11	DE	2014.1	3	1,246.06
12	DE	2014.1	4	449.01
13	FR	2014.1	1	69,361.23
14	FR	2014.1	2	9,005.76
15	FR	2014.1	3	1,919.15
16	FR	2014.1	4	226.09
17	GB	2014.1	1	66,549.83
18	GB	2014.1	2	3,898.64
19	GB	2014.1	3	168.87
20	GB	2014.1	4	40.37
21	US	2014.1	1	986,354.35
22	US	2014.1	2	141,250.16
23	US	2014.1	3	17,109.73
24	US	2014.1	4	3,955.67
25	AU	2014.1	1	130,620.49
26	AU	2014.1	2	14,405.04
27	AU	2014.1	3	2,960.4
28	AU	2014.1	4	753.67
29	CA	2014.1	1	465,975.34
30	CA	2014.1	2	52,642.88
31	CA	2014.1	3	5,233.09
32	CA	2014.1	4	1,311.94
33	DE	2014.1	1	36,045.17
34	DE	2014.1	2	8,376.45
35	DE	2014.1	3	1,246.06
36	DE	2014.1	4	449.01
37	FR	2014.1	1	69,361.23
38	FR	2014.1	2	9,005.76
39	FR	2014.1	3	1,919.15
40	FR	2014.1	4	226.09
41	GB	2014.1	1	66,549.83
42	GB	2014.1	2	3,898.64
43	GB	2014.1	3	168.87
44	GB	2014.1	4	40.37
45	US	2014.1	1	986,354.35
46	US	2014.1	2	141,250.16
47	US	2014.1	3	17,109.73
48	US	2014.1	4	3,955.67

Figure 11. Data loaded into “plan_data” table (48 rows).

quarterid	status	modifieddatetime	author	country
2014.1	R	2022-08-19 11:08:15.247	ivan	AU
2014.1	R	2022-08-19 11:08:15.247	ivan	CA
2014.1	R	2022-08-19 11:08:15.247	ivan	DE
2014.1	R	2022-08-19 11:08:15.247	ivan	FR
2014.1	R	2022-08-19 11:08:15.247	ivan	GB
2014.1	R	2022-08-19 11:08:15.247	ivan	US

Figure 12. Data loaded into “plan_status” table (6 rows).

Task №7. Changing plan data.

```
def set_lock(year, quarter, user, pwd):
    con = psycopg2.connect(database='2022_plans_alexey_nepochatov',
                           user=user,
                           password=pwd,
                           host='localhost')

    cur = con.cursor()
    year_quarter = str(year) + '.' + str(quarter)

    try:
        cur.execute(f"""
            update plan_status as ps
            set status = 'L',
                modifieddatetime = now(),
                author = current_user
            where ps.quarterid = {year_quarter}::text
                and ps.status = 'R'
                and country in (
                    select cm.country
                    from country_managers as cm
                    where cm.username = current_user)

        """)

    finally:
        con.commit()
        con.close()

def remove_lock(year, quarter, user, pwd):
    con = psycopg2.connect(database='2022_plans_alexey_nepochatov',
                           user=user,
                           password=pwd,
                           host='localhost')

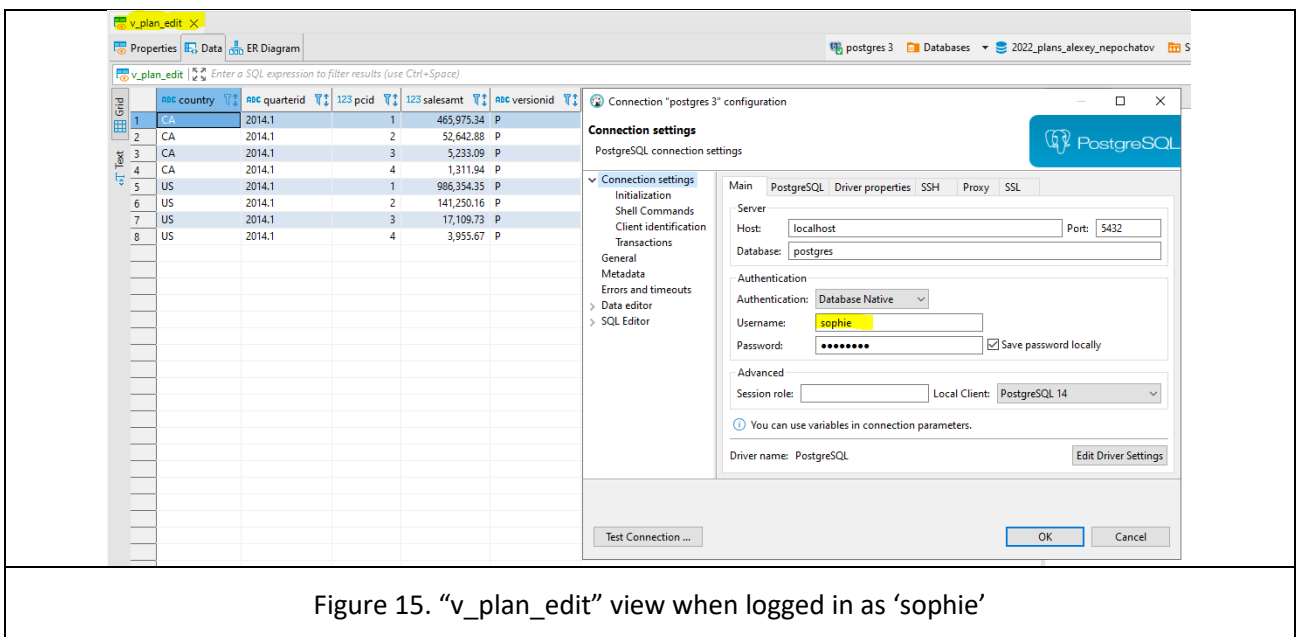
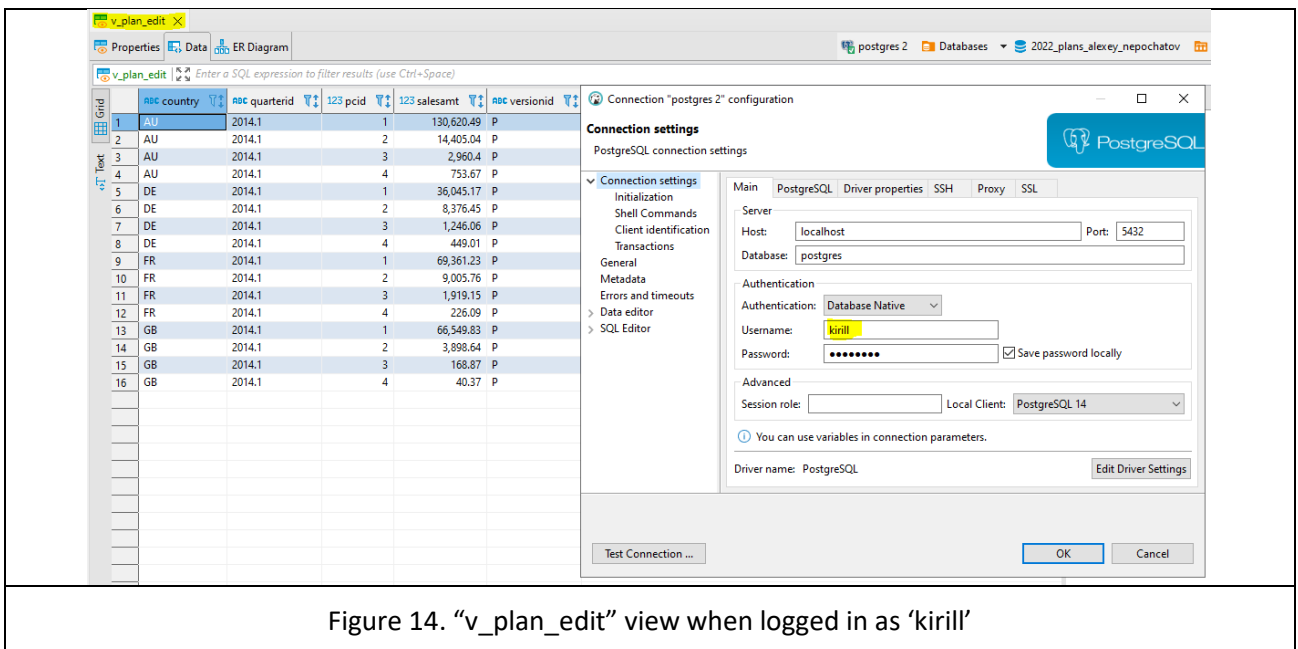
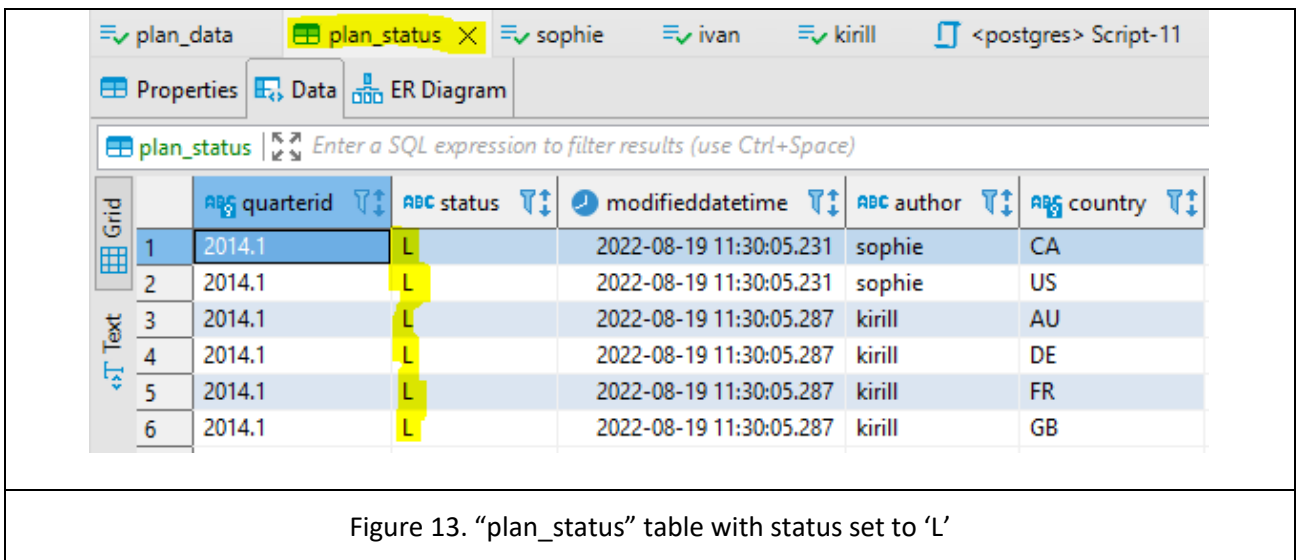
    cur = con.cursor()
    year_quarter = str(year) + '.' + str(quarter)

    try:
        cur.execute(f"""
            update plan_status as ps
            set status = 'R',
                modifieddatetime = now(),
                author = current_user
            where ps.quarterid = {year_quarter}::text
                and ps.status = 'L'
                and country in (
                    select cm.country
                    from country_managers as cm
                    where cm.username = current_user)

        """)

    finally:
        con.commit()
        con.close()

if __name__ == '__main__':
    set_lock(2014, 1, 'sophie', 'password')
    set_lock(2014, 1, 'kirill', 'password')
```



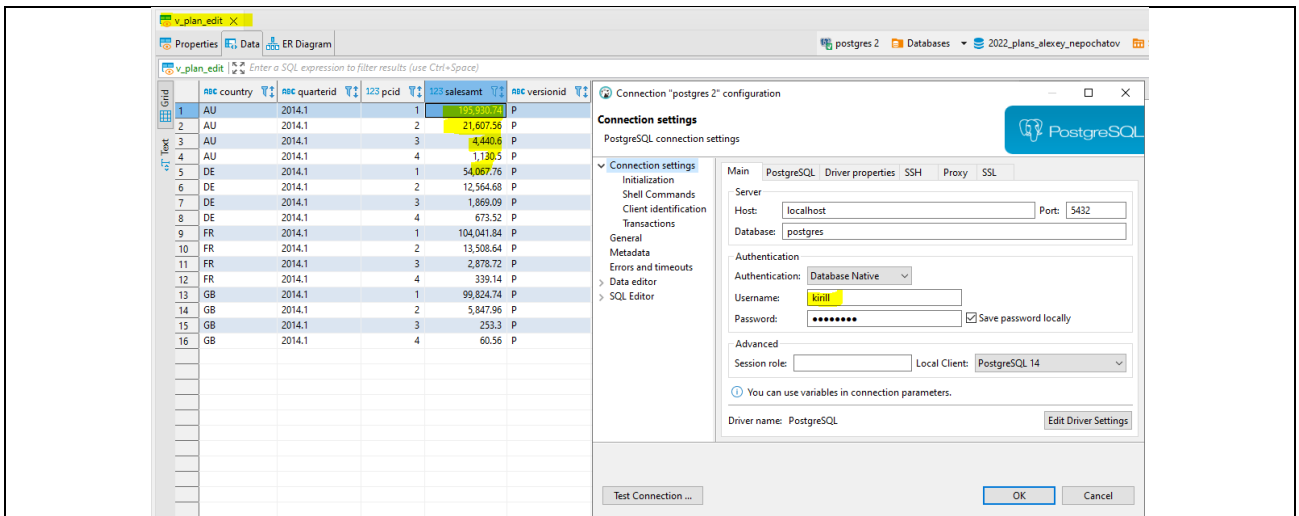


Figure 16. Increasing 'saleamt' by 50% when logged in as 'kirill'

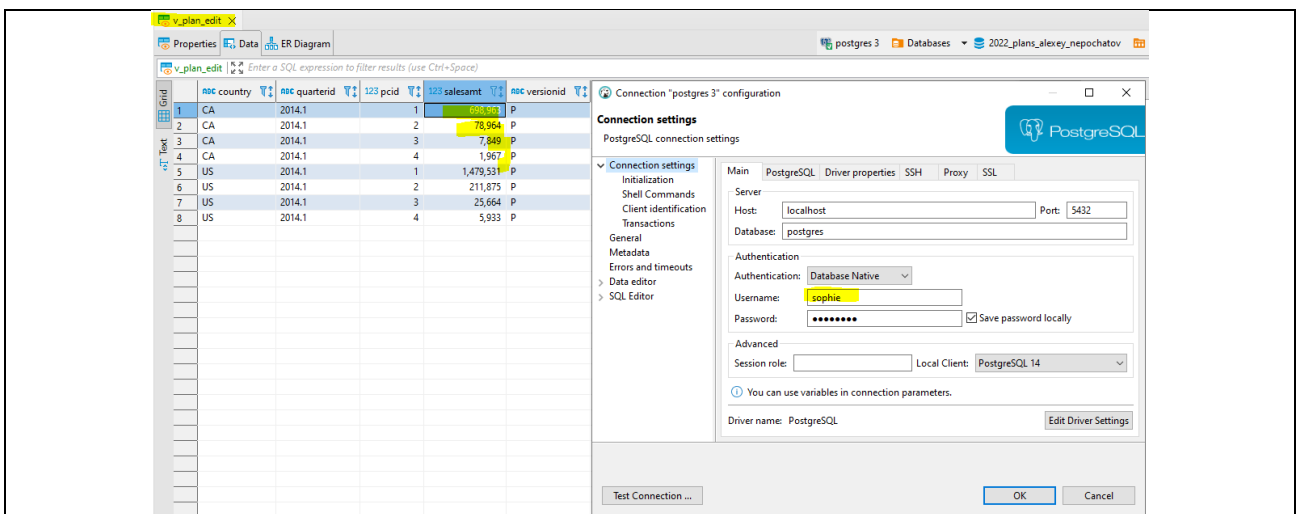


Figure 17. Increasing 'saleamt' by 50% when logged in as 'sophie'

Executing the "remove_lock" function

```
if __name__ == '__main__':
    remove_lock(2014, 1, 'kirill', 'password')
    remove_lock(2014, 1, 'sophie', 'password')
```

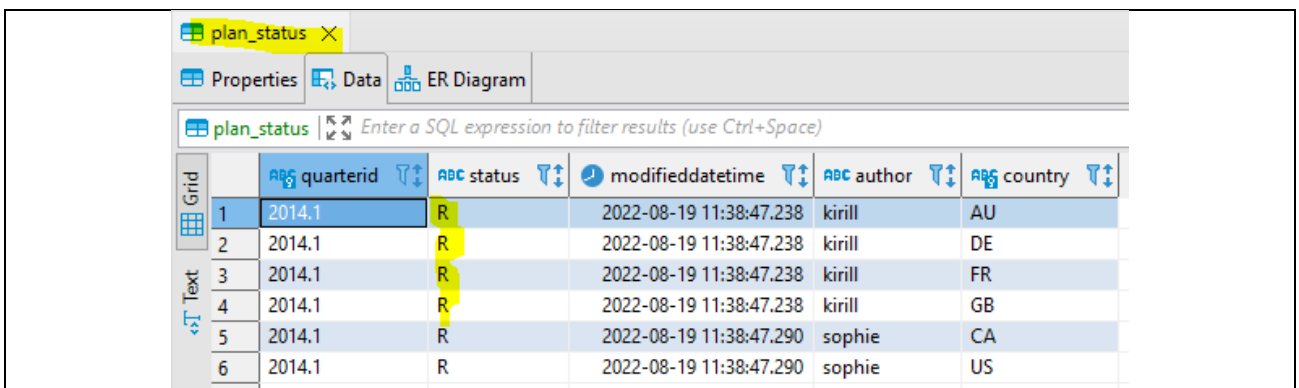


Figure 18. "plan_status" table with status set to 'R' after the "remove_lock" function was executed.

Task №8. Plan data approval.

```
def accept_plan(year, quarter, user, pwd=''):
    con = psycopg2.connect(database='2022_plans_alexey_nepochatov',
                           user=user,
                           password=pwd,
                           host='localhost')

    cur = con.cursor()
    year_quarter = str(year) + '.' + str(quarter)
    try:
        # Clear the A version of plan data
        cur.execute(f"""
            delete from plan_data as pd
            where pd.quarterid = {year_quarter}::text
                and pd.versionid = 'A'
                and pd.country in (
                    select cm.country
                    from country_managers as cm
                    where cm.username = current_user)
        """)
        # Read data and save its copy into as the version A
        cur.execute(f"""
            insert into plan_data
            select 'A' as versionid,
                pd.country as country,
                pd.quarterid as quarterid,
                pd.pcid as pcid,
                pd.salesamt as salesamt
            from plan_data as pd
            where pd.quarterid = {year_quarter}::text
                and pd.versionid = 'P'
                and pd.country in (
                    select cm.country
                    from country_managers as cm
                    where cm.username=current_user)
            and pd.country in (
                select ps.country
                from plan_status as ps
                where ps.quarterid = {year_quarter}::text
                    and ps.status = 'R')
        """)
        # Change the status of data being processed from 'R' to 'A'
        cur.execute(f"""
            update plan_status as ps
            set status = 'A',
                modifieddatetime = now(),
                author = current_user
            where ps.quarterid = {year_quarter}::text
                and ps.status = 'R'
                and country in (
                    select cm.country
                    from country_managers as cm
                    where cm.username = current_user)
        """)

    finally:
        con.commit()
        con.close()

if __name__ == '__main__':
    accept_plan(2014, 1, 'kirill', 'password')
    accept_plan(2014, 1, 'sophie', 'password')
```

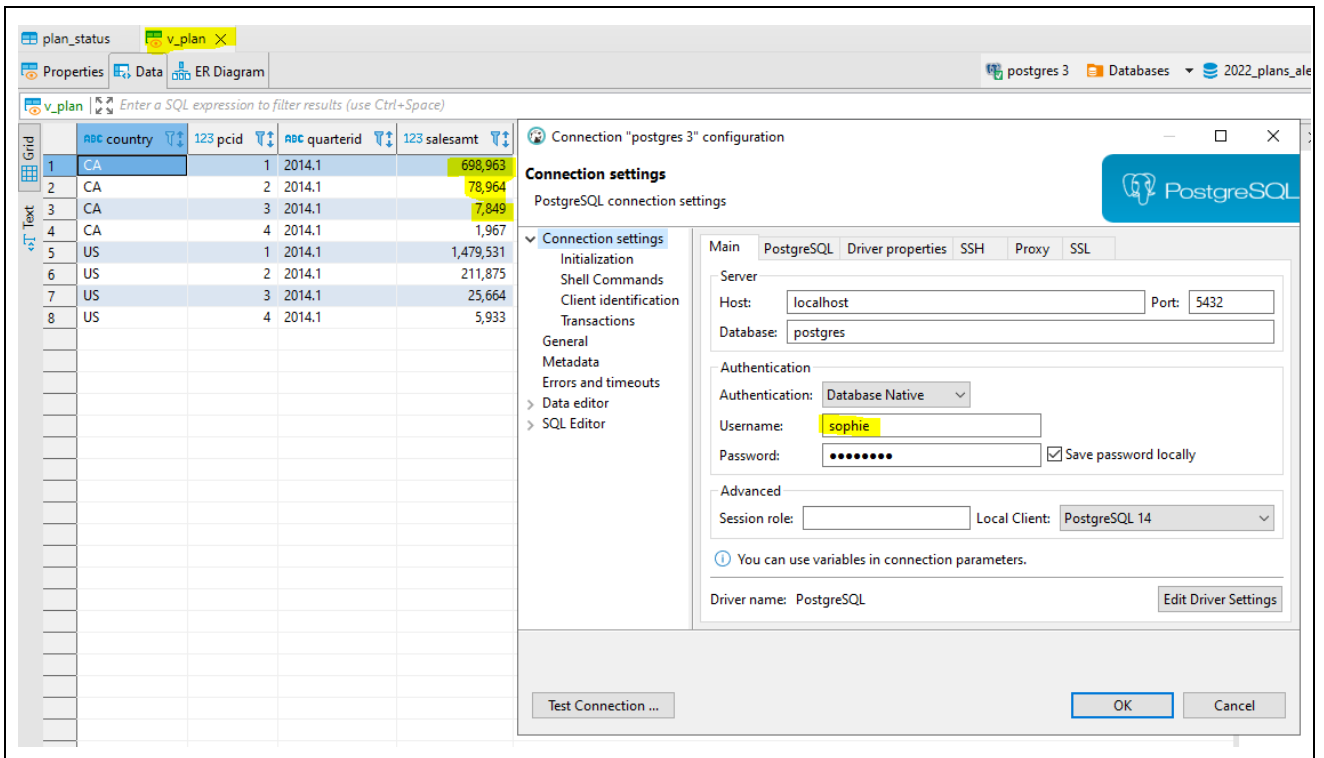


Figure 19. "v_plan" view when logged in as 'sophie' after the "accept_plan" function was executed.

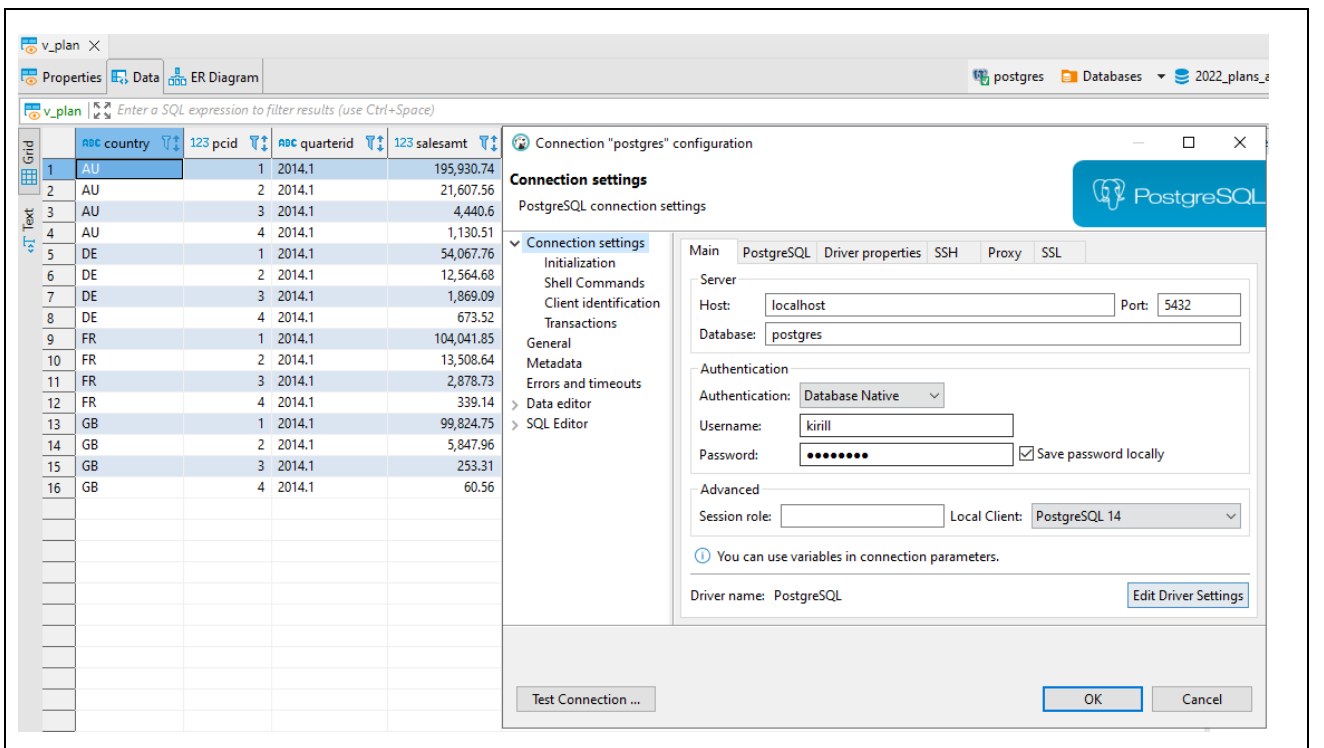


Figure 20. "v_plan" view when logged in as 'kirill' after the "accept_plan" function was executed.

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

```
create materialized view mv_plan_fact_2014_q1 as
select
  quarterid as "Quarter",
  country as "Country",
  category_name as "Category name",
  plan.salesamt - fact.salesamt as "Dev.",
  round((plan.salesamt - fact.salesamt)/plan.salesamt * 100, 1) as "Dev., %"
from
  (select
    c.countrycode as country,
    p2.pcid,
    p2.pcname as category_name,
    sum(sod.linetotal) as salesamt
  from
    salesorderheader as soh
    join customer as cst
      using (customerid)
    join salesorderdetail as sod
      using(salesorderid)
    join product2 as p2
      using (productid)
    join company as c
      on c.cname = cst.companyname
  where extract('year' from soh.orderdate) in('2014')
    and extract('quarter' from soh.orderdate) in('1')
    and c.id in (
      select abc.cid
      from company_abc as abc
      where abc."year" in('2013')
        and abc.cls in('A', 'B'))
  group by c.countrycode,
    p2.pcid,
    p2.pcname) as fact
join
  (select * from plan_data as pd
   where pd.versionid = 'A') as plan
  using(country)
  where plan.pcid = fact.pcid
with no data;

refresh materialized view mv_plan_fact_2014_q1;
```

mv_plan_fact_2014_q1 X					
Properties Data ER Diagram					
mv_plan_fact_2014_q1 Enter a SQL expression to filter results (use Ctrl+Space)					
Grid	ABC Quarter	ABC Country	ABC Category name	123 Dev.	123 Dev., %
1	2014.1	AU	Bikes	-31,023.562	-15.8
2	2014.1	AU	Components	-3,007.398	-13.9
3	2014.1	AU	Clothing	-1,622.3672	-36.5
4	2014.1	AU	Accessories	-2,536.09	-224.3
5	2014.1	CA	Bikes	439,501.49	62.9
6	2014.1	CA	Components	37,626.868	47.7
7	2014.1	CA	Clothing	-2,710.016943	-34.5
8	2014.1	CA	Accessories	-1,623.052	-82.5
9	2014.1	DE	Bikes	-17,503.228	-32.4
10	2014.1	DE	Components	4,010.444	31.9
11	2014.1	DE	Clothing	-976.582868	-52.2
12	2014.1	DE	Accessories	-830.822411	-123.4
13	2014.1	FR	Bikes	52,001.084	50
14	2014.1	FR	Components	5,067.402	37.5
15	2014.1	FR	Clothing	1,730.761892	60.1
16	2014.1	US	Bikes	35,003.18412	2.4
17	2014.1	US	Components	-82,904.843088	-39.1
18	2014.1	US	Clothing	-15,413.426805	-60.1
19	2014.1	US	Accessories	-11,858.951696	-199.9

Figure 21. Materialized view “mv_plan_fact_2014_q1” data

Dump database:

CMD
command:

cd C:\Program Files\PostgreSQL\14\bin

pg_dump -h localhost -U postgres -F c -b -v -f "C:\Users\Alexey\Desktop\HSE\01
Programming\06 SQL\Week 6\Final
Task\Final_Task_Nepochatov\2022_plans_alexey_nepochatov.dump"
2022_plans_alexey_nepochatov

```

Командная строка
pg_dump: выгрузка содержимого таблицы "public.address"
pg_dump: выгрузка содержимого таблицы "public.company"
pg_dump: выгрузка содержимого таблицы "public.company_abc"
pg_dump: выгрузка содержимого таблицы "public.company_sales"
pg_dump: выгрузка содержимого таблицы "public.country_managers"
pg_dump: выгрузка содержимого таблицы "public.customer"
pg_dump: выгрузка содержимого таблицы "public.customeraddress"
pg_dump: выгрузка содержимого таблицы "public.plan_data"
pg_dump: выгрузка содержимого таблицы "public.plan_status"
pg_dump: выгрузка содержимого таблицы "public.product"
pg_dump: выгрузка содержимого таблицы "public.productcategory"
pg_dump: выгрузка содержимого таблицы "public.productdescription"
pg_dump: выгрузка содержимого таблицы "public.productphoto"
pg_dump: выгрузка содержимого таблицы "public.productproductphoto"
pg_dump: выгрузка содержимого таблицы "public.productssubcategory"
pg_dump: выгрузка содержимого таблицы "public.salesorderdetail"
pg_dump: выгрузка содержимого таблицы "public.salesorderheader"

C:\Program Files\PostgreSQL\14\bin>

```