```
/* DDL file */

CREATE TABLE plan_status (
quarterid varchar(6) NOT NULL,
status varchar(10) NOT NULL,
modifieddatetime timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
author varchar(20) NOT NULL DEFAULT CURRENT_USER,
country varchar(5) NOT NULL,
CONSTRAINT plan_status_pk PRIMARY KEY (quarterid, country)
);

CREATE TABLE plan_data (
versionid varchar(1) NOT NULL,
country varchar(5) NOT NULL,
quarterid varchar(6) NOT NULL,
pcid int4 NOT NULL,
salesamt numeric(18,2) NULL,
CONSTRAINT planapp_data_pkey PRIMARY KEY (quarterid, country, pcid, versionid)
);

CREATE TABLE country_managers (
username varchar(30) NOT NULL,
country varchar(5) NOT NULL,
CONSTRAINT country_managers_pk PRIMARY KEY (username, country)
);

CREATE TABLE company (
id int4 NOT NULL GENERATED ALWAYS AS IDENTITY,
cname varchar(200) NOT NULL,
countrycode varchar(10) NULL,
city varchar(30) NULL,
CONSTRAINT d_company_pk PRIMARY KEY (id)
);

CREATE TABLE company_sales (
cid int4 NOT NULL,
salesamt numeric(18,2) NULL,
year int4 NULL,
quarter_yr int4 NULL,
qr varchar(6) NOT NULL,
categoryid int4 NOT NULL,
ccls varchar(1) NULL,
CONSTRAINT company_sales_pk PRIMARY KEY (qr, cid, categoryid)
);

CREATE TABLE company_abc (
cid int4 NOT NULL,
salestotal numeric NULL,
cls varchar(1) NULL,
year int4 NOT NULL,
CONSTRAINT company_abc_pk PRIMARY KEY (cid, year)
);

create view v_plan_edit as
select pd.country, pd.quarterid, pd.pcid, pd.salesamt, pd.versionid
from plan_data pd
where
```

```
pd.versionid = 'P'
and
pd.country in (select country
from country_managers cm
where cm.username = current_user)
and
pd.quarterid in (select ps.quarterid
from plan_status ps
where ps.author = current_user and ps.status = 'L');

create view v_plan as
select pd.country,
pd.pcid,
pd.quarterid,
pd.salesamt
FROM plan_data pd
WHERE pd.versionid = 'A'
AND (pd.country IN (SELECT cm.country FROM country_managers cm
WHERE cm.username = CURRENT_USER)
or
pg_has_role(current_user, 'planadmin', 'member'))
AND (pd.quarterid IN ( SELECT ps.quarterid
FROM plan_status ps
WHERE ps.status = 'A'));

create role planadmin;

create role planmanager;
```

/* Task №1. Access settings */

```
grant select on all tables in schema public to planadmin, planmanager;

grant select, update, insert, delete on plan_data to planadmin;
grant select, update, insert, delete on plan_status to planadmin;
grant select, update, insert, delete on country_managers to planadmin;
revoke all privileges on v_plan_edit from planadmin;
revoke all privileges on v_plan from planadmin;

grant select, update, insert, delete on plan_data to planmanager;
grant select, update on plan_status to planmanager;
grant select on country_managers to planmanager;
grant select, update on v_plan_edit to planmanager;
grant select on v_plan to planmanager;

create user ivan with password 'sql1';
grant planadmin to ivan;

create user sophie with password 'sql2';
grant planmanager to sophie;

create user kirill with password 'sql3';
grant planmanager to kirill


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 */

create materialized view product2 as
select p2.productcategoryid as pcid, p.productid,
         p2.name as pcname, p.name as pname
from product as p
join productsubcategory p2 on p.productsubcategoryid = p2.productsubcategoryid

create materialized view country2 as
select distinct countryregioncode as countrycode from address
join (select * from customeraddress where addresstype = 'Main Office') as c
on address.addressid = c.addressid

grant select on product2, country2 to planadmin, planmanager;



/* Task №3. Loading data into the company table */

insert into company(cname, countrycode, city)
select c.companyname as cname,
         a.countryregioncode as countrycode, a.city
from customer 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 */

```
insert into company_abc (cid, salestotal, cls, year)
select  cid, salestotal,
            CASE
            WHEN sum(salestotal) over (partition by b.year order by salestotal desc) <=
0.80*s THEN 'A'
            WHEN sum(salestotal) over (partition by b.year order by salestotal desc)
between 0.80*s and 0.95*s THEN 'B'
            ELSE 'C'
          END AS cls,
            b.year as year
from
(select c.id as cid,
         sum(soh.subtotal) as salestotal,
       date_part('y', soh.orderdate) as year
from salesorderheader as soh
join customer as cs on cs.customerid = soh.customerid
join company as c on c.cname = cs.companyname
where date_part('y', soh.orderdate) in (2012, 2013)
group by c.id, date_part('y', soh.orderdate)
order by date_part('y', soh.orderdate), sum(soh.subtotal) desc) as b
join
(select date_part('y', orderdate) as year,
         sum(subtotal) as S
from salesorderheader as soh
join customer as cs on cs.customerid = soh.customerid
join company as c on c.cname = cs.companyname
where date_part('y', orderdate) in (2012, 2013)
group by date_part('y', orderdate)) as d on b.year = d.year
```

| | cid | salestotal | cls | year |
|---|---|---|---|---|
| 1 | 116 | 375,493.4641 | A | 2,012 |
| 2 | 146 | 351,188.4604 | A | 2,012 |
| 3 | 25 | 316,681.8038 | A | 2,012 |
| 4 | 32 | 301,678.2118 | A | 2,012 |
| 5 | 193 | 296,800.7702 | A | 2,012 |
| 6 | 66 | 289,303.2579 | A | 2,012 |
| 7 | 9 | 274,221.0413 | A | 2,012 |
| 8 | 49 | 265,936.5862 | A | 2,012 |
| 9 | 38 | 263,035.9455 | A | 2,012 |
| 10 | 42 | 219,829.2882 | A | 2,012 |
| 11 | 56 | 213,869.4374 | A | 2,012 |
| 12 | 133 | 202,777.6034 | A | 2,012 |
| 13 | 46 | 190,732.7335 | A | 2,012 |
| 14 | 51 | 186,628.455 | A | 2,012 |
| 15 | 147 | 174,683.8141 | A | 2,012 |
| 16 | 54 | 172,701.4457 | A | 2,012 |
| 17 | 28 | 166,732.7648 | A | 2,012 |
| 18 | 145 | 164,883.5653 | A | 2,012 |
| 19 | 85 | 154,657.3032 | A | 2,012 |
| 20 | 77 | 152,685.4219 | A | 2,012 |
| 21 | 69 | 151,136.7341 | A | 2,012 |
| 22 | 22 | 150,166.369 | A | 2,012 |
| 23 | 23 | 148,670.6653 | A | 2,012 |
| 24 | 121 | 144,520.7627 | A | 2,012 |
| 25 | 126 | 142,557.8217 | A | 2,012 |
| 26 | 20 | 140,109.3983 | A | 2,012 |
| 27 | 127 | 137,631.3086 | A | 2,012 |

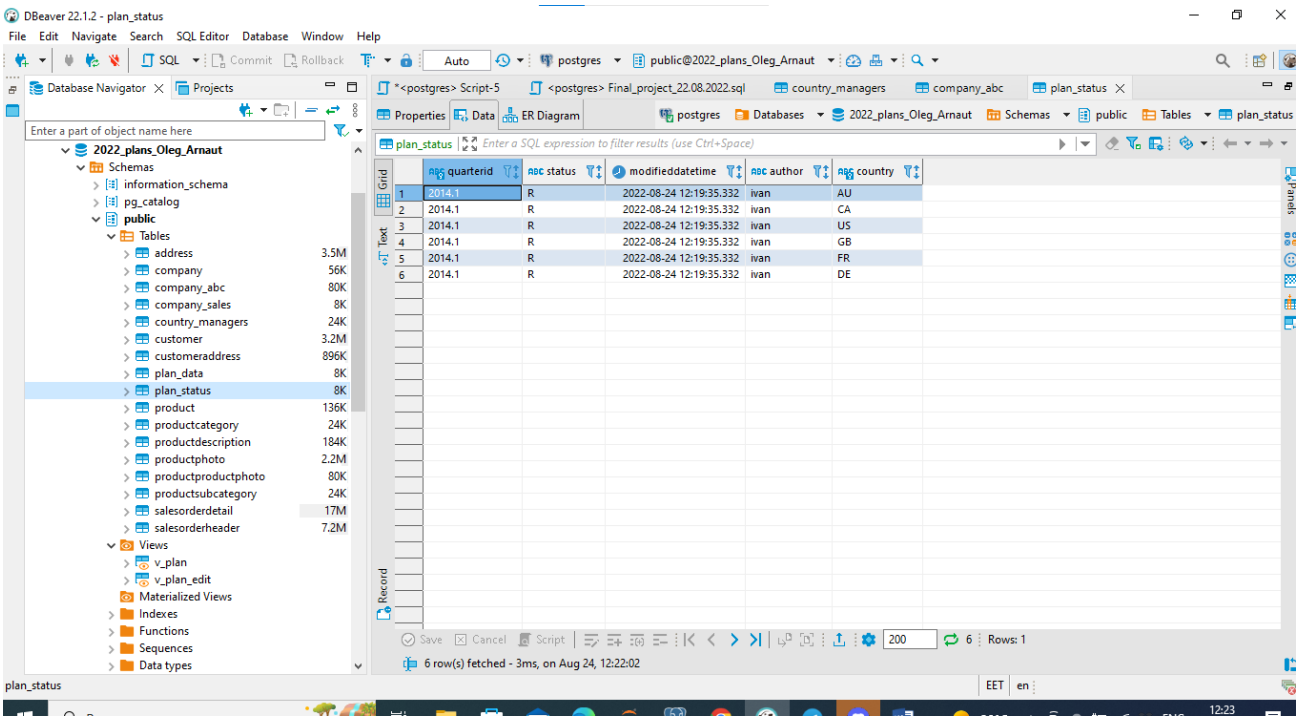
/* Task №5 Findind quarterlysales amount by company and product category */

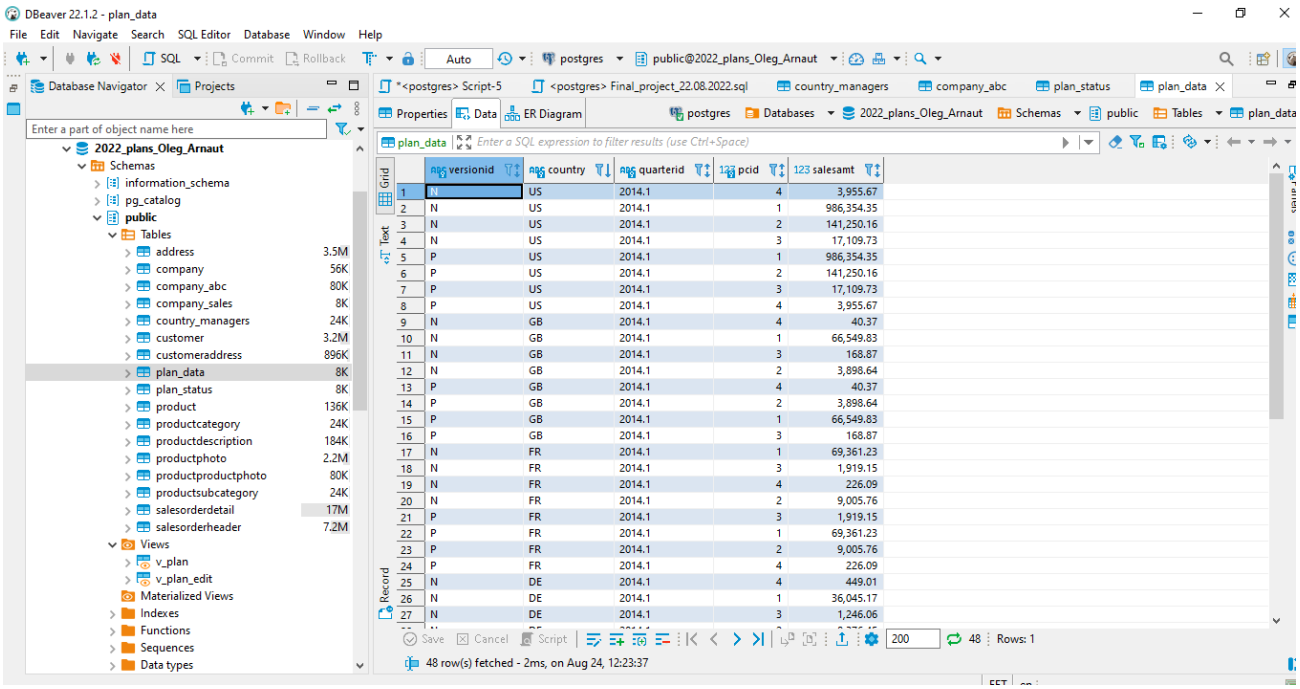
```
insert into company_sales(cid, salesamt, year, quarter_yr, qr, categoryid, ccls)
select c.id as cid, sum(s.linetotal),
         date_part('y', soh.orderdate) as year,
         date_part('quarter', soh.orderdate) as quarter_yr,
         concat_ws('.', date_part('y', soh.orderdate), date_part('quarter',
soh.orderdate)) as qr,
         p.pcid as categoryid, ca.cls as ccls
from company c
join company_abc ca on c.id = ca.cid
join (select * from customer where customerid > 1000) as cs on cs.companyname = c.cname
join salesorderheader as soh on soh.customerid = cs.customerid and date_part('y',
soh.orderdate) = ca.year
join salesorderdetail s on soh.salesorderid = s.salesorderid
join product2 p on p.productid = s.productid
group by categoryid, c.id, date_part('y', soh.orderdate), date_part('quarter',
soh.orderdate), ccls
order by categoryid;
```

/* Task №6. Initial data preparation */

```
import psycopg2
def start_planning(year, quarter, user, pwd):

    con = psycopg2.connect(database=" 2022_plans_Oleg_Arnaut", user=user, password=pwd,
host="localhost")
    cur = con.cursor()
    cur.execute(f"delete from plan_status where quarterid = cast({year}.{quarter} as
varchar)")
    cur.execute(f"delete from plan_data where quarterid = cast({year}.{quarter} as
varchar)")
    for country in ['AU', 'CA', 'US', 'GB', 'FR', 'DE']:
        cur.execute(f"insert into plan_status(quarterid, status, country)\
                    values('{year}.{quarter}', 'R', '{country}');")

    for cat in ('N', 'P'):
        cur.execute(f"""insert into plan_data(versionid, country, quarterid, pcid,
salesamt)
                       select versionid, country, quarterid, pcid, salesamt
                       from
                       (select distinct pcid, countrycode as country, '{cat}'as
versionid,
                               '{year}.{quarter}' as quarterid, 0 as salesamt
                       from product2 p2 cross join country2 c2) as d """)

    for cat in ('N', 'P'):
        cur.execute(f"""update plan_data
                        set salesamt = c.avg
                        from
                        (select * from
                        (select versionid, country, '{year}.{quarter}' as quarterid,
pcid, avg(salesamt)
                        from (select 'N' as versionid, c.countrycode as country,
                        cs.qr as quartedid, cs.categoryid as pcid,
```

```
                        sum(cs.salesamt) as salesamt
                        from company_sales cs
                        join company c on cs.cid = c.id
                        where cs.ccls in ('A', 'B') and cs.quarter_yr = {quarter}
                        group by country, cs.qr, pcid
                        order by country, pcid, quartedid) as b
                        group by b.versionid, b.country, b.pcid) as c) as c
                      where plan_data.country = c.country and plan_data.pcid = c.pcid
""")

    con.commit()
    con.close()


user, pwd, year, quarter   = 'ivan', 'sql1', 2014, 1
start_planning(year, quarter, user, pwd)
```

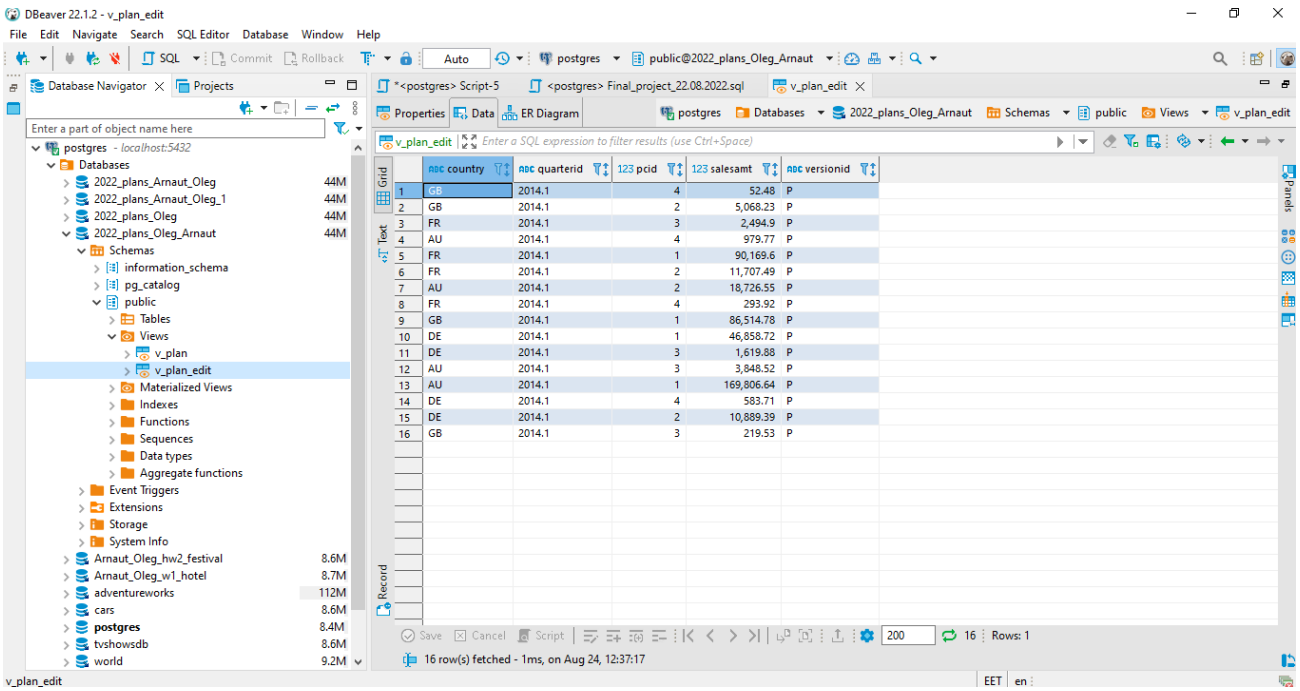





## /* Task №7. Changing plan data */

```
/* set_lock */

def set_lock(year, quarter, user, pwd):

    con = psycopg2.connect(database="2022_plans_Oleg_Arnaut", user=user, password=pwd,
host="localhost")
    cur = con.cursor()

    cur.execute(f"""update plan_status
                    set status = 'L',  modifieddatetime = CURRENT_TIMESTAMP, author =
(SELECT CURRENT_USER)
                    where plan_status.country in (select country from country_managers
where username = (SELECT CURRENT_USER)
                    and plan_status.quarterid = '{year}.{quarter}')
                    """)
    con.commit()
    con.close()

user = 'kirill'
pwd = 'sql3'
year = 2014
quarter = 1
set_lock(year, quarter, user, pwd)

user = 'sophie'
pwd = 'sql2'
year = 2014
quarter = 1
set_lock(year, quarter, user, pwd)

/* update, increasing by 30% and 50% */
```

```
def update(year, quarter, user, pwd, inc):
    con = psycopg2.connect(database="2022_plans_Oleg_Arnaut", user=user, password=pwd,
host="localhost")
    cur = con.cursor()
    cur.execute(f"""update v_plan_edit
                    set salesamt = salesamt + {inc}*salesamt
                    where quarterid = '{year}.{quarter}' and
                    country in (select country from country_managers where username =
'{user}') """)
    con.commit()
    con.close()

user = 'kirill'
pwd = 'sql3'
year = 2014
quarter = 1
inc = 0.3
update(year, quarter, user, pwd, inc)

user = 'sophie'
pwd = 'sql2'
year = 2014
quarter = 1
inc = 0.5
update(year, quarter, user, pwd, inc)
```




```
/* remove_lock */

def remove_lock(year, quarter, user, pwd):

    con = psycopg2.connect(database="2022_plans_Oleg_Arnaut", user=user, password=pwd,
host="localhost")
    cur = con.cursor()
```

```
    cur.execute(f"""update plan_status
                    set status = 'R',  modifieddatetime = CURRENT_TIMESTAMP, author =
(SELECT CURRENT_USER)
                    where plan_status.country in (select country from country_managers
where username = (SELECT CURRENT_USER)
                    and plan_status.quarterid = '{year}.{quarter}')
                    """)
    con.commit()
    con.close()

user = 'kirill'
pwd = 'sql3'
year = 2014
quarter = 1
remove_lock(year, quarter, user, pwd)

user = 'sophie'
pwd = 'sql2'
year = 2014
quarter = 1
remove_lock(year, quarter, user, pwd
```

## /* Task №8. Plan data approval */

Note !!! According to the primary settings (Task 1) the planadmin does not have any permissions for v_plan

*Table 2 Planning system users' rights*

| | User role | |
|---|---|---|
| **DB object** | planadmin | planmanager |
| all tables | S | S |
| plan_data | SUID | SUID |
| plan_status | SUID | SU |
| country_managers | SUID | S |
| v_plan_edit | - | SU |
| v_plan | - | S |

This, in turn, has a contradiction with Task 8 conditions, where is mentioned that administrator (planadmin) has access for entire plan:

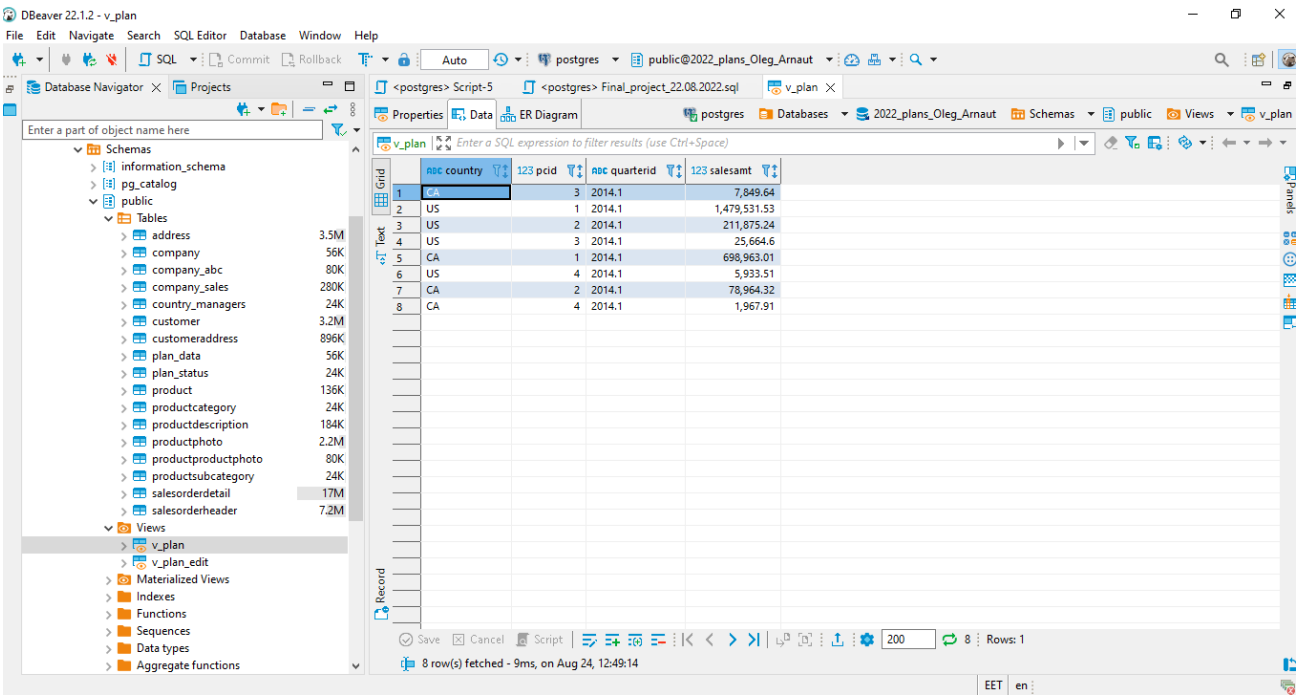
When updating the status, also save a time stamp in *modifiedtimestamp* column. Use the developed function to approve the plan of Q1 2014 on behalf of each manager. Check whether the data is visible through the *v_plan* view:

- The administrator has access to the entire plan
- Manager can view only data he/she is permitted to read and change.

So, I have change the permissions for planadmin in order to solve this problem

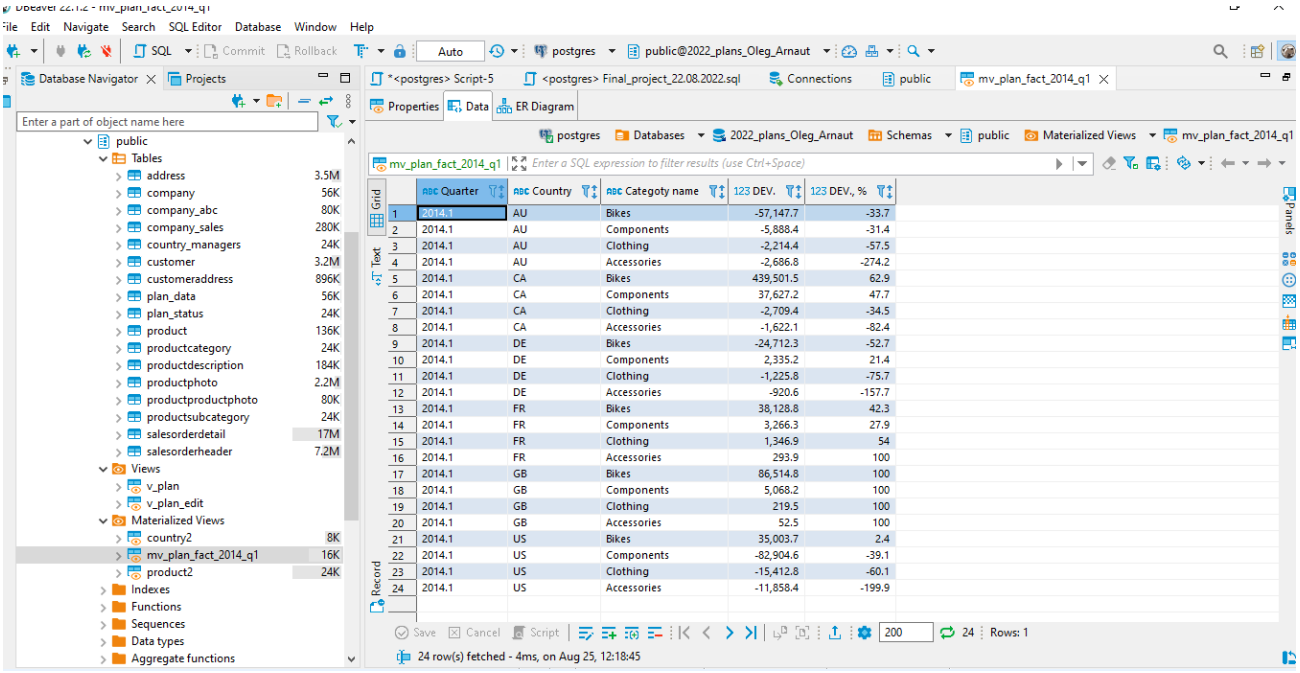
```
grant select on v_plan to planadmin;
```

```
def accept_plan(year, quarter, user, pwd):

    con = psycopg2.connect(database="2022_plans_Oleg_Arnaut", user=user, password=pwd,
host="localhost")
    cur = con.cursor()
    cur.execute(f"""delete from plan_data where quarterid = '{year}.{quarter}'
                                                and versionid = 'A'
                                                and country in (select country from
country_managers where username = '{user}') """)



    cur.execute(f"""insert into plan_data(versionid, country, quarterid, pcid, salesamt)
                     select 'A', b.country, quarterid, pcid, salesamt
                     from(select * from plan_data where versionid = 'P' and quarterid =
'{year}.{quarter}' ) as b
                     join country_managers cm on b.country = cm.country
                     where cm.username = '{user}'""")


    cur.execute(f"""update plan_status
                     set status = 'A',  modifieddatetime = CURRENT_TIMESTAMP
                     where plan_status.country in (select country from country_managers
where username = '{user}'
                     and plan_status.quarterid = '{year}.{quarter}')
                     """)

    con.commit()
    con.close()


user = 'kirill'
pwd = 'sql3'
year = 2014
quarter = 1
accept_plan(year, quarter, user, pwd)

user = 'sophie'
pwd = 'sql2'
year = 2014
quarter = 1
accept_plan(year, quarter, user, pwd)
```




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

```sql
--The second option (actual data estimation using salesorderheader and ordersalesdetail
--tables without using company_sales) was applied.

create materialized view mv_plan_fact_2014_q1 as
select quarterid as "Quarter",
       country as "Country",
       p."name" as "Categoty name",
       --plan.salesamt as plan,
       --COALESCE(fact, 0) as fact,
       round(salesamt - COALESCE(fact, 0), 1) as "DEV.",
       round((salesamt - COALESCE(fact, 0))*100/salesamt, 1) as "DEV., %"
from v_plan as plan
join productcategory as p on  p.productcategoryid = plan.pcid
left join (select co.countrycode,   sum(sod.linetotal) as fact,  p2.pcid
from company as co
        join customer as cs on co.cname = cs.companyname
        join salesorderheader as soh using(customerid)
        join salesorderdetail as sod using(salesorderid)
        join product2 as p2 using(productid)
where
        date_part('y', soh.orderdate) = '2014'
        and date_part('quarter', soh.orderdate) = 1
        and co.id in (select co2.cid
                      from company_abc as co2
                      where co2.year = '2013' and co2.cls in ('A', 'B'))
group by co.countrycode, p2.pcid) as actual
on plan.country = actual.countrycode and plan.pcid = actual.pcid
order by country;
```




| | Quarter | Country | Categoty name | DEV. | DEV., % |
|---|---|---|---|---|---|
| 1 | 2014.1 | AU | Bikes | -57,147.7 | -33.7 |
| 2 | 2014.1 | AU | Components | -5,888.4 | -31.4 |
| 3 | 2014.1 | AU | Clothing | -2,214.4 | -57.5 |
| 4 | 2014.1 | AU | Accessories | -2,686.8 | -274.2 |
| 5 | 2014.1 | CA | Bikes | 439,501.5 | 62.9 |
| 6 | 2014.1 | CA | Components | 37,627.2 | 47.7 |
| 7 | 2014.1 | CA | Clothing | -2,709.4 | -34.5 |
| 8 | 2014.1 | CA | Accessories | -1,622.1 | -82.4 |
| 9 | 2014.1 | DE | Bikes | -24,712.3 | -52.7 |
| 10 | 2014.1 | DE | Components | 2,335.2 | 21.4 |
| 11 | 2014.1 | DE | Clothing | -1,225.8 | -75.7 |
| 12 | 2014.1 | DE | Accessories | -920.6 | -157.7 |
| 13 | 2014.1 | FR | Bikes | 38,128.8 | 42.3 |
| 14 | 2014.1 | FR | Components | 3,266.3 | 27.9 |
| 15 | 2014.1 | FR | Clothing | 1,346.9 | 54 |
| 16 | 2014.1 | FR | Accessories | 293.9 | 100 |
| 17 | 2014.1 | GB | Bikes | 86,514.8 | 100 |
| 18 | 2014.1 | GB | Components | 5,068.2 | 100 |
| 19 | 2014.1 | GB | Clothing | 219.5 | 100 |
| 20 | 2014.1 | GB | Accessories | 52.5 | 100 |
| 21 | 2014.1 | US | Bikes | 35,003.7 | 2.4 |
| 22 | 2014.1 | US | Components | -82,904.6 | -39.1 |
| 23 | 2014.1 | US | Clothing | -15,412.8 | -60.1 |
| 24 | 2014.1 | US | Accessories | -11,858.4 | -199.9 |