# Task №1. Access settings

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

grant select, update, insert, delete on plan\_data, plan\_status, country\_managers to 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;

create user sophie;

create user kirill;

grant planadmin to ivan;

grant planmanager to sophie, kirill;

select

\*

from information\_schema.role\_table\_grants where grantee = 'planadmin';

insert into country\_managers

values

('sophie', 'US'), ('sophie', 'CA'),

('kirill', 'FR'), ('kirill', 'GB'), ('kirill', 'DE'), ('kirill', 'AU');

# Task №2. product2 & country 2 materialized views

create materialized view product2 as

select

pc.productcategoryid as pcid,

p.productid as productid,

pc.*name* as pcname,

p.*name* as pname

from product p

join productcategory pc on pc.productcategoryid = p.productsubcategoryid;

select \* from product2;

create materialized view country2 as

select distinct countryregioncode

from address;

grant select on product2 to planadmin, planmanager;

grant select on country2 to planadmin, planmanager;

# Task №3. Loading data into the company table

insert into company (cname, countrycode, city)

select

c.companyname,

a.countryregioncode,

a.city

from customer c

join customeraddress ca on ca.customerid = c.customerid

join address a on ca.addressid = a.addressid

where c.companyname is not null;

# Task №4. Company classification

insert into company\_abc

select

customerid as cid,

st as salestotal,

CASE

WHEN srt <= (select 0.8 \* *sum*(soh.subtotal) as s\_a

from customer c

join salesorderheader soh on soh.customerid = c.customerid

where c.companyname is not null and year = *date\_part*('y', soh.orderdate)) THEN 'A'

WHEN srt <= (select 0.95 \* *sum*(soh.subtotal) as s\_b

from customer c

join salesorderheader soh on soh.customerid = c.customerid

where c.companyname is not null and year = *date\_part*('y', soh.orderdate)) THEN 'B'

ELSE 'C'

END cls,

year

from

(select

customerid,

companyname,

year,

st,

*sum*(st) over (partition by year rows between unbounded preceding and current row) srt

from (

select c.customerid, companyname, *date\_part*('y', soh.orderdate) as year, *sum*(soh.subtotal) as st

from customer c

join salesorderheader soh on soh.customerid = c.customerid

where c.companyname is not null

group by c.customerid, year

order by st desc) as data

where year in ('2012', '2013')) as data2;

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

insert into company\_sales

select

cid,

salesamt,

year,

quarter as quarter\_yr,

year || '.' || quarter as qr,

categoryid,

cls as ccls

from(

select

c.customerid as cid,

*sum*(sod.linetotal) as salesamt,

*date\_part*('y', soh.orderdate) as year,

*date\_part*('quarter', soh.orderdate) as quarter,

p.pcid as categoryid

from customer c

join salesorderheader soh on soh.customerid = c.customerid

join salesorderdetail sod on soh.salesorderid = sod.salesorderid

join product2 p on sod.productid = p.productid

where c.companyname is not null

group by c.customerid, year, quarter, p.pcid) as d1

join company\_abc using (cid, year);

# Task №6. Initial data preparation

def start\_planning(year, quarter, user, pwd):

con = psycopg2.connect(

database='y2022\_plans\_yumen', user=user, password=pwd, host='localhost')

quarter\_id = year + '.' + quarter

cur = con.cursor()

'''

1. Delete plan data from the plan\_data table related to the target year and quarter.

'''

cur.execute(f'delete from plan\_data where quarterid=\'{quarter\_id}\'')

'''

In the plan\_status table delete records related to the target quarter

'''

cur.execute(f'delete from plan\_status where quarterid=\'{quarter\_id}\'')

'''

2. Create planning status records (plan\_status table) for the selected quarter. The number of records added equals the number of countries in which customer-companies (shops) are situated.

'''

cur.execute(

f'''

insert into plan\_status (quarterid, status, country)

select

distinct

{quarter\_id} as quarterid,

'R' as status,

co.countrycode as country

from company\_sales cs

join customer cu on cs.cid = cu.customerid

join company co on co.cname = cu.companyname;

''')

'''

3. Generate version N of planning data in the plan\_data table. Use the calculation algorithm is described in section 1.4. on the page.

'''

cur.execute(

f'''

insert into plan\_data

select

'N' as versionid,

country,

'{year}.' || quarter\_yr as quarterid,

categoryid as pcid,

avg(salesamt)

from

(select

cs.quarter\_yr as quarter\_yr,

cs.categoryid as categoryid,

co.countrycode as country,

sum(salesamt) as salesamt

from

company\_sales cs

join customer cu on cs.cid = cu.customerid

join company co on co.cname = cu.companyname

where cs.ccls in ('A', 'B') and quarter\_yr = {quarter}

group by cs.year, cs.quarter\_yr, cs.categoryid, co.countrycode) as sum\_comp

group by sum\_comp.quarter\_yr, sum\_comp.categoryid, sum\_comp.country;

''')

'''

4. Copy data from version N into version P in the plan\_data table.

'''

cur.execute(

f'''

insert into plan\_data

select

'P' as versionid,

country,

quarterid,

pcid,

salesamt

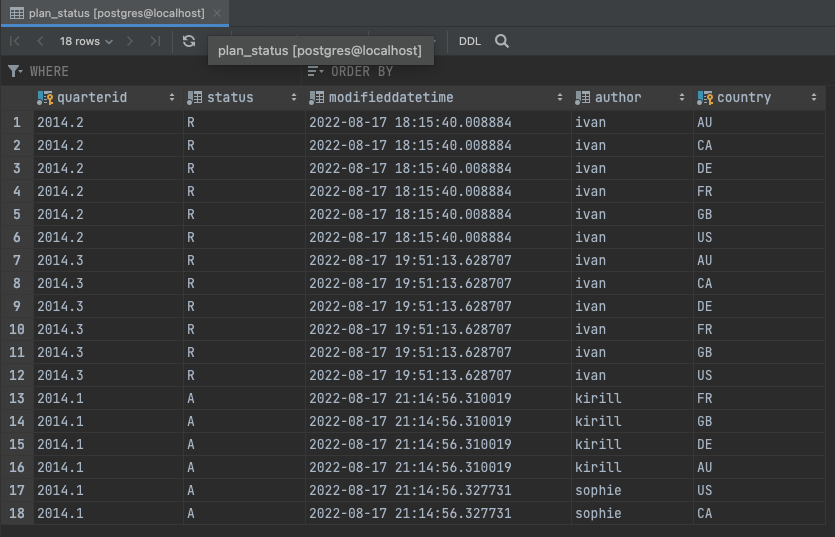
from plan\_data

where versionid = 'N' and quarterid = '{quarter\_id}';

''')

con.commit()

start\_planning('2014', '1', 'ivan', None)



## Image

## Changing plan data

def change\_lock(year, quarter, user, pwd, status):

con = psycopg2.connect(

database='y2022\_plans\_yumen', user=user, password=pwd, host='localhost')

quarter\_id = year + '.' + quarter

cur = con.cursor()

cur.execute(

f'''

update plan\_status

set status = '{status}',

author = current\_user,

modifieddatetime = current\_timestamp

where country in (select country from country\_managers where username = current\_user)

and quarterid = '{quarter\_id}'

''')

con.commit()

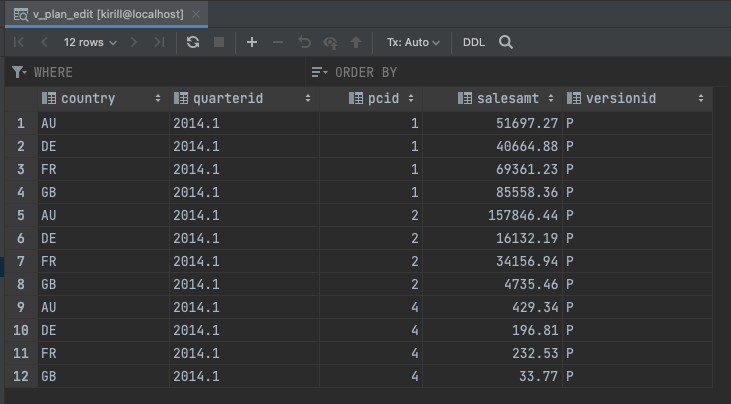
def set\_lock(year, quarter, user, pwd):

change\_lock(year, quarter, user, pwd, 'L')

def remove\_lock(year, quarter, user, pwd):

change\_lock(year, quarter, user, pwd, 'R')

set\_lock('2014', '1', 'kirill', None)

set\_lock('2014', '1', 'sophie', None)

## Plan data approval

def accept\_plan(year, quarter, user, pwd):

con = psycopg2.connect(

database='y2022\_plans\_yumen', user=user, password=pwd, host='localhost')

quarter\_id = year + '.' + quarter

cur = con.cursor()

cur.execute(f'''

delete from plan\_data

where quarterid = '{quarter\_id}'

and versionid = 'A'

and country in (select country from country\_managers where username = current\_user)

''')

cur.execute(f'''

insert into plan\_data

select

'A' as status,

pd.country as country,

pd.quarterid as quarterid,

pd.pcid as pcid,

pd.salesamt as salesamt

from plan\_data pd

left join plan\_status ps on ps.quarterid = pd.quarterid and ps.country = pd.country

left join country\_managers cm on pd.country = cm.country

where pd.quarterid = '{quarter\_id}'

and pd.versionid = 'P'

and ps.status = 'R'

and cm.username = current\_user

''')

cur.execute(

f'''

update plan\_status

set status = 'A',

author = current\_user,

modifieddatetime = current\_timestamp

where country in (select country from country\_managers where username = current\_user)

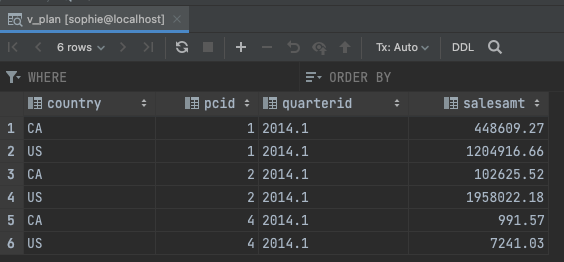
and quarterid = '{quarter\_id}'

''')

con.commit()

accept\_plan('2014', '1', 'kirill', None)

accept\_plan('2014', '1', 'sophie', None)



# Data preparation for plan-fact analysis in Q1 2014

I chose approch 1 and loaded data of 2014 into the company\_sales table and include this table in the view.

create materialized view mv\_plan\_fact\_2014\_q1 as

select

plan.quarterid as quater,

plan.country as country,

categoryname,

plan.salesamt - fact.salesamt as dev,

(plan.salesamt - fact.salesamt) / plan.salesamt as dev\_perc

from

(select

year || '.' || quarter\_yr as quarterid,

countrycode,

categoryid,

name as categoryname,

salesamt

from (

select

year,

quarter\_yr,

co.countrycode,

cs.categoryid,

pc.name,

*sum*(cs.salesamt) as salesamt

from

company\_sales cs

join customer cu on cs.cid = cu.customerid

join company co on co.cname = cu.companyname

join productcategory pc on pc.productcategoryid = cs.categoryid

where year = '2013' and ccls in ('A', 'B')

group by year, quarter\_yr, co.countrycode, cs.categoryid, pc.name) as data

where quarter\_yr = 1) as fact

join (select

country,

quarterid,

pcid,

salesamt

from plan\_data

where quarterid = '2014.1' and versionid = 'A') as plan

on fact.categoryid = plan.pcid and fact.countrycode = plan.country

