***Task №1. Access settings***

**-- Set up permissions for roles as written in the table below**

**--**Legend:

**--**S – select

**--**U – update

**--**I – insert

**--**D – delete

**--Role planadmin**

**--S on all tables**

**GRANT SELECT ON TABLE** public.address **TO** planadmin;

**GRANT SELECT ON TABLE** public.company **TO** planadmin;

**GRANT SELECT ON TABLE** public.company\_abc **TO** planadmin;

**GRANT SELECT ON TABLE** public.company\_sales **TO** planadmin;

**GRANT SELECT ON TABLE** public.country\_managers **TO** planadmin;

**GRANT SELECT ON TABLE** public.customer **TO** planadmin;

**GRANT SELECT ON TABLE** public.customeraddress **TO** planadmin;

**GRANT SELECT ON TABLE** public.plan\_data **TO** planadmin;

**GRANT SELECT ON TABLE** public.plan\_status **TO** planadmin;

**GRANT SELECT ON TABLE** public.product **TO** planadmin;

**GRANT SELECT ON TABLE** public.productcategory **TO** planadmin;

**GRANT SELECT ON TABLE** public.productsubcategory **TO** planadmin;

**GRANT SELECT ON TABLE** public.salesorderdetail **TO** planadmin;

**GRANT SELECT ON TABLE** public.salesorderheader **TO** planadmin;

**--SUID on plan\_data**  **GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE** public.plan\_data **TO** planadmin;

**--SUID on plan\_status  
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE** public.plan\_status **TO** planadmin;  
  
**--SUID on country\_managers  
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE** public.country\_managers **TO** planadmin;

**--Role planmanager**

**--S on all tables**

**GRANT SELECT ON TABLE** public.address **TO** planmanager;

**GRANT SELECT ON TABLE** public.company **TO** planmanager;

**GRANT SELECT ON TABLE** public.company\_abc **TO** planmanager;

**GRANT SELECT ON TABLE** public.company\_sales **TO** planmanager;

**GRANT SELECT ON TABLE** public.country\_managers **TO** planmanager;

**GRANT SELECT ON TABLE** public.customer **TO** planmanager;

**GRANT SELECT ON TABLE** public.customeraddress **TO** planmanager;

**GRANT SELECT ON TABLE** public.plan\_data **TO** planmanager;

**GRANT SELECT ON TABLE** public.plan\_status **TO** planmanager;

**GRANT SELECT ON TABLE** public.product **TO** planmanager;

**GRANT SELECT ON TABLE** public.productcategory **TO** planmanager;

**GRANT SELECT ON TABLE** public.productsubcategory **TO** planmanager;

**GRANT SELECT ON TABLE** public.salesorderdetail **TO** planmanager;

**GRANT SELECT ON TABLE** public.salesorderheader **TO** planmanager;

**--SUID on plan\_data**  **GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE** public.plan\_data **TO** planmanager;

**--SU on plan\_status  
GRANT SELECT, UPDATE ON TABLE** public.plan\_status **TO** planmanager;  
  
**--S on country\_managers  
GRANT SELECT ON TABLE** public.country\_managers **TO** planmanager;  
 **--SU on v\_plan\_edit  
GRANT SELECT, UPDATE ON TABLE** public.v\_plan\_edit **TO** planmanager;

**--S on v\_plan  
GRANT SELECT ON TABLE** public.v\_plan **TO** planmanager;

**--Create users**

**CREATE USER** ivan with **PASSWORD** '111111'; **CREATE USER** sophie with **PASSWORD** '222222'; **CREATE USER** kirill with **PASSWORD** '333333';

**--Grant roles to users**

**GRANT** planadmin **TO** ivan;  
**GRANT** planmanager **TO** sophie;  
**GRANT** planmanager **TO** kirill;

**--Insert values**

**INSERT INTO** public.country\_managers (username,country)

**VALUES**

('sophie','US'),

('sophie','CA'),

('kirill','FR'),

('kirill','GB'),

('kirill','DE'),

('kirill','AU');

***Task №2. product2 & country 2 materialized views***

**--Create materialized view product2**

**CREATE MATERIALIZED VIEW** product2 **AS**  
**SELECT**  
productcategory.productcategoryid **AS** pcid,  
product.productid **AS** productid,  
productcategory.name **AS** pcname,  
product.name **AS** pname  
**FROM**  
(  
**SELECT**  
product.productsubcategoryid,  
product.productid,  
product.name  
**FROM** product) **AS** product  
**LEFT JOIN** productsubcategory **ON** product.productsubcategoryid = productsubcategory.productsubcategoryid  
**LEFT JOIN** productcategory **ON** productsubcategory.productcategoryid = productcategory.productcategoryid  
;

**--Create materialized view country2**

**CREATE MATERIALIZED VIEW** country2 **AS**

**SELECT DISTINCT**

address.countryregioncode **AS** countrycode

**FROM**

address

;

**--Grant permissions**

**GRANT SELECT ON** product2, country2 **TO** planadmin, planmanager;

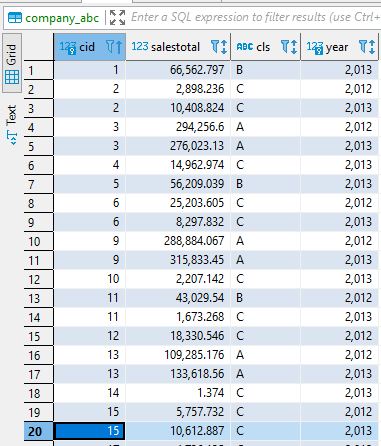
***Task №3. Loading data into the company table***

**INSERT INTO** company(cname, countrycode, city)

**SELECT**  
customer.companyname **AS** cname,  
max(address.countryregioncode) **AS** countrycode,  
max(address.city) **AS** city  
**FROM**  
(  
**SELECT**  
customerid,  
companyname  
**FROM** customer  
) **AS** customer  
**INNER JOIN** customeraddress **ON** customer.customerid = customeraddress.customerid  
**INNER JOIN** address **ON** customeraddress.addressid = address.addressid  
**WHERE**  
customeraddress.addresstype='Main Office'  
**GROUP BY** customer.companyname  
;

***Task №4. Company classification***

**INSERT INTO** company\_abc(cid, salestotal, cls, year) **SELECT**  
cid,  
salestotal,  
**CASE**  
 **WHEN**  
 **SUM**(salestotal) **OVER** (**PARTITION BY** year **ORDER** **BY** salestotal desc)/  
 **SUM**(salestotal) **OVER** (**PARTITION BY** year) \*100 <=80  
 **THEN** 'A'  
 **WHEN**   
 **SUM**(salestotal) **OVER** (**PARTITION** **BY** year **ORDER BY** salestotal desc)/  
 **SUM**(salestotal) **OVER** (**PARTITION** **BY** year) \*100 <=95  
 **THEN** 'B'  
 **ELSE** 'C'  
**END AS** cls,  
year  
**FROM**  
(  
**SELECT**  
salesorderheader.year **AS** year,  
company.id **AS** cid,  
**SUM**(salesorderheader.subtotal) **AS** salestotal   
**FROM**  
(  
**SELECT**  
date\_part('year',orderdate) **AS** year,  
customerid,  
subtotal  
**FROM** salesorderheader) **AS** salesorderheader  
**INNER JOIN** customer **ON** salesorderheader.customerid = customer.customerid  
**INNER JOIN** company **ON** customer.companyname = company.cname  
**WHERE**  
customer.companyname **IS NOT NULL AND**  
(  
year = 2012 **OR**  
year = 2013  
)  
**GROUP BY**  
company.id, salesorderheader.year  
**ORDER BY**  
year **ASC**,salestotal **DESC**  
) **AS** company\_abc  
**ORDER BY** cid **ASC**, year **ASC**  
;



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

**INSERT INTO** company\_sales(cid, salesamt, year, quarter\_yr, qr, categoryid, ccls) **SELECT**  
company.id **AS** cid,  
**SUM**(salesorderdetail.linetotal) **AS** salesamt,  
date\_part('year',salesorderheader.orderdate) **AS** year,  
date\_part('quarter',salesorderheader.orderdate) **AS** quarter\_yr,  
concat(date\_part('year',salesorderheader.orderdate),'.',date\_part('quarter',salesorderheader.orderdate)) **AS** qr,  
product2.pcid **AS** categoryid,  
company\_abc.cls **AS** ccls  
**FROM**  
(  
**SELECT**  
salesorderid,  
productid,  
linetotal  
**FROM** salesorderdetail) **AS** salesorderdetail  
**LEFT JOIN** product2 **ON** salesorderdetail.productid = product2.productid  
**INNER JOIN** salesorderheader **ON** salesorderdetail.salesorderid = salesorderheader.salesorderid  
**INNER JOIN** customer **ON** salesorderheader.customerid = customer.customerid  
**INNER JOIN** company **ON** customer.companyname = company.cname  
**INNER JOIN** company\_abc **ON** company.id = company\_abc.cid **AND** date\_part('year',salesorderheader.orderdate) = company\_abc.year  
**GROUP BY** company.id,date\_part('year',salesorderheader.orderdate),date\_part('quarter',salesorderheader.orderdate),product2.pcid,company\_abc.cls  
**ORDER BY** cid **ASC**, qr **ASC**, categoryid **ASC**   
;

***Task №6. Initial data preparation***

import psycopg2

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

    connection = psycopg2.connect(database="2021\_plans\_DavidSchmid", user=user, password=pwd, host="localhost")

    cursor = connection.cursor()

    #1. Delete plan data from the plan\_data table related to the target year and quarter. In the plan\_status table delete records related to the target quarter

    cursor.execute(f"""

    DELETE FROM plan\_data

    WHERE quarterid='{year}.{quarter}'

    """)

    cursor.execute(f"""

    DELETE FROM plan\_status

    WHERE quarterid='{year}.{quarter}'

    """)

    #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.

    cursor.execute(f"""

    INSERT INTO plan\_status(quarterid, country, status, modifieddatetime, author)

    SELECT DISTINCT

    '{year}.{quarter}' AS quarterid,

    countrycode AS country,

    'R' AS status,

    CURRENT\_TIMESTAMP AS modifieddatetime,

    current\_user AS author

    FROM country2

    """)

    #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.

    cursor.execute(f"""

    INSERT INTO plan\_data(versionid, country, quarterid, pcid, salesamt)

    SELECT

    'N' as versionid,

    company.countrycode AS country,

    '{year}.{quarter}' AS quarterid,

    company\_sales.pcid,

    SUM(company\_sales.salesamt)/2 as salesamt

    FROM

    (

    SELECT

    cid,

    qr AS quarterid,

    categoryid AS pcid,

    salesamt

    FROM company\_sales

    WHERE ccls = 'A' or ccls = 'B') AS company\_sales

    INNER JOIN company ON company\_sales.cid = company.id

    WHERE

    (

    split\_part(quarterid,'.',1)::INTEGER = {year-1} OR

    split\_part(quarterid,'.',1)::INTEGER = {year-2}

    )

    AND

    (

    split\_part(quarterid,'.',2)::INTEGER = {quarter}

    )

    GROUP BY company\_sales.pcid, company.countrycode

    """)

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

    cursor.execute(f"""

    INSERT INTO plan\_data(versionid, country, quarterid, pcid, salesamt)

    SELECT

    'P' as versionid,

    country,

    quarterid,

    pcid,

    salesamt

    FROM

    plan\_data

    WHERE

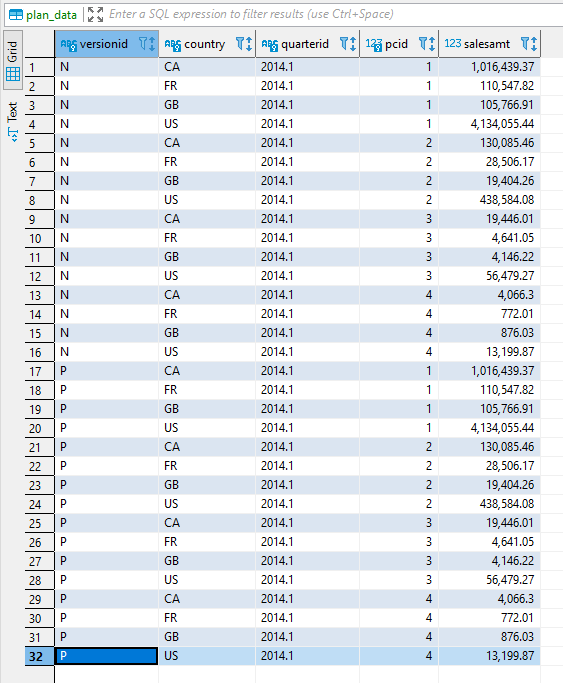
    quarterid = '{year}.{quarter}'

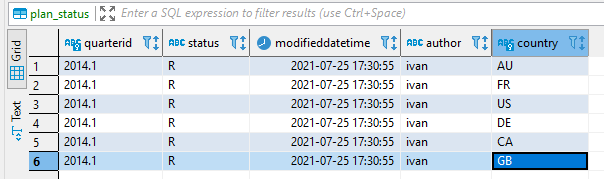
    """)

    connection.commit()

    connection.close()

start\_planning(2014, 1, 'ivan', '111111')





***Task №7. Changing plan data***

import psycopg2

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

    connection = psycopg2.connect(database="2021\_plans\_DavidSchmid", user=user, password=pwd, host="localhost")

    cursor = connection.cursor()

    #1. Delete plan data from the plan\_data table related to the target year and quarter. In the plan\_status table delete records related to the target quarter

    cursor.execute(f"""

    UPDATE plan\_status

    SET

    status = 'L',

    modifieddatetime = CURRENT\_TIMESTAMP,

    author = CURRENT\_USER

    FROM country\_managers

    WHERE

    country\_managers.country = plan\_status.country AND

    country\_managers.username = '{user}' AND

    plan\_status.quarterid = '{year}.{quarter}'

    """)

    connection.commit()

    connection.close()

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

    connection = psycopg2.connect(database="2021\_plans\_DavidSchmid", user=user, password=pwd, host="localhost")

    cursor = connection.cursor()

    #1. Delete plan data from the plan\_data table related to the target year and quarter. In the plan\_status table delete records related to the target quarter

    cursor.execute(f"""

    UPDATE plan\_status

    SET

    status = 'R',

    modifieddatetime = CURRENT\_TIMESTAMP,

    author = CURRENT\_USER

    FROM country\_managers

    WHERE

    country\_managers.country = plan\_status.country AND

    country\_managers.username = '{user}' AND

    plan\_status.quarterid = '{year}.{quarter}'

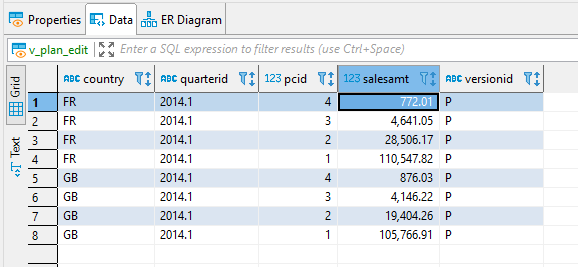
    """)

    connection.commit()

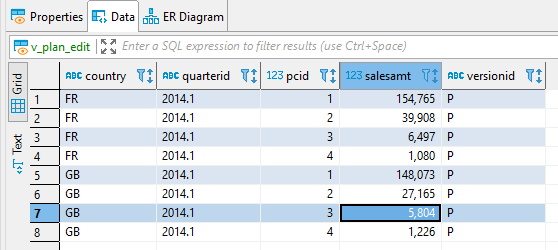
    connection.close()

set\_lock(2014, 1, "kirill", "333333")

#remove\_lock(2014, 1, "kirill", "333333")

Before change:  


After change +40%:



***Task №8. Plan data approval***

import psycopg2

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

    connection = psycopg2.connect(database="2021\_plans\_DavidSchmid", user=user, password=pwd, host="localhost")

    cursor = connection.cursor()

    cursor.execute(f"""

    UPDATE plan\_status

    SET

    status = 'A',

    modifieddatetime = CURRENT\_TIMESTAMP,

    author = CURRENT\_USER

    FROM country\_managers

    WHERE

    country\_managers.country = plan\_status.country AND

    country\_managers.username = '{user}' AND

    plan\_status.quarterid = '{year}.{quarter}'

    """)

    cursor.execute(f"""

    DELETE

    FROM plan\_data

    USING country\_managers

    WHERE

    plan\_data.country = country\_managers.country AND

    plan\_data.quarterid = '{year}.{quarter}' AND

    country\_managers.username = '{user}' AND

    plan\_data.versionid = 'A'

    """)

    cursor.execute(f"""

    INSERT INTO plan\_data(versionid, country, quarterid, pcid, salesamt)

    SELECT

    'A' AS versionid,

    plan\_data.country,

    plan\_data.quarterid,

    plan\_data.pcid,

    plan\_data.salesamt

    FROM country\_managers,plan\_data

    WHERE

    plan\_data.country = country\_managers.country AND

    plan\_data.quarterid = '{year}.{quarter}' AND

    country\_managers.username = '{user}' AND

    plan\_data.versionid = 'P'

    """)

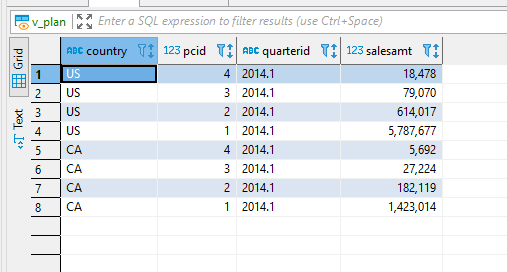
    connection.commit()

    connection.close()

accept\_plan(2014, 1, "sophie", "222222")

accept\_plan(2014, 1, "kirill", "333333")

Logged in as Sophie, rows of v\_plan:



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

**Decision of approach:**

I decided to use the approach to **“Load data of 2014 into the company\_sales table and include this table in the view”**

Therefore, I do 2 steps:

**1. Load data of 2014 into the company\_sales table**:

**INSERT INTO** company\_sales(cid, salesamt, year, quarter\_yr, qr, categoryid, ccls) **SELECT**

company.id **AS** cid,

**SUM**(salesorderdetail.linetotal) **AS** salesamt,

date\_part('year',salesorderheader.orderdate) **AS** year,

date\_part('quarter',salesorderheader.orderdate) **AS** quarter\_yr,

concat(date\_part('year',salesorderheader.orderdate),'.',date\_part('quarter',salesorderheader.orderdate)) **AS** qr,

product2.pcid **AS** categoryid,

company\_abc.cls **AS** ccls

**FROM**

(

**SELECT**

salesorderid,

productid,

linetotal

**FROM** salesorderdetail) **AS** salesorderdetail

**LEFT JOIN** product2 **ON** salesorderdetail.productid = product2.productid

**INNER JOIN** salesorderheader **ON** salesorderdetail.salesorderid = salesorderheader.salesorderid

**INNER JOIN** customer **ON** salesorderheader.customerid = customer.customerid

**INNER JOIN** company **ON** customer.companyname = company.cname

**INNER JOIN** company\_abc **ON** company.id = company\_abc.cid **AND** company\_abc.year = '2013'

**WHERE** date\_part('year',salesorderheader.orderdate) = '2014'

**GROUP BY** company.id,date\_part('year',salesorderheader.orderdate),date\_part('quarter',salesorderheader.orderdate),product2.pcid,company\_abc.cls

**ORDER BY** cid **ASC**, qr **ASC**, categoryid **ASC**

;

**2. Include company sales (with other necessary transformations) into the materialized view mv\_plan\_fact\_2014\_q1:**

**CREATE MATERIALIZED VIEW** mv\_plan\_fact\_2014\_q1 AS  
**SELECT**  
quarter,  
country,  
category\_name,  
**CASE**  
 **WHEN**

**SUM**(salesamt\_plan)=0  
 **THEN**  
 NULL  
 **ELSE**  
 **SUM**(salesamt\_plan) - **SUM**(salesamt\_act)  
**END AS** dev\_amount,  
**CASE**  
 **WHEN**  
 **SUM**(salesamt\_plan)=0  
 **THEN**  
 NULL  
 **ELSE**  
 concat(round((**SUM**(salesamt\_plan) - **SUM**(salesamt\_act))/ **SUM**(salesamt\_plan)\*100,3)::Text,' %')  
**END AS** deviation\_rel  
**FROM**  
(  
(  
**SELECT**  
company\_sales.quarter,  
company.countrycode **AS** country,  
product2.pcname **AS** category\_name,  
**SUM**(company\_sales.salesamt) as salesamt\_act,  
0 **AS** salesamt\_plan  
**FROM**  
(

**SELECT**cid,qr **AS** quarter,categoryid **AS** pcid,salesamt **FROM** company\_sales **WHERE**qr='2014.1' **AND**(ccls = 'A' or ccls = 'B')) **AS** company\_sales **INNER JOIN** company **ON** company\_sales.cid = company.id **LEFT JOIN**( **SELECT DISTINCT**pcid,pcname **FROM** product2) **AS** product2 ON company\_sales.pcid = product2.pcid **GROUP BY** company\_sales.quarter, company.countrycode, product2.pcname) **UNION**( **SELECT**plan\_data.quarter,plan\_data.country,product2.pcname **AS** category\_name,0 **AS** salesamt\_act,salesamt **AS** salesamt\_plan **FROM**( **SELECT**pcid,quarterid **AS** quarter,country,salesamt  **FROM** plan\_data **WHERE** versionid='A') **AS** plan\_data **LEFT JOIN**( **SELECT DISTINCT**pcid,pcname **FROM** product2) **AS** product2 ON plan\_data.pcid = product2.pcid)) **AS** mv\_plan\_fact\_2014\_q1  
**GROUP BY** country,category\_name,quarter  
**ORDER BY** country,category\_name,quarter;

