SQL Final Project

Aliaksandr Lemiasheuski

August 2023

Contents

[Task №1. A query/queries to put the data into the ‘country\_managers’ table 3](#_Toc142242206)

[Task №2. product2 & country 2 materialized views 3](#_Toc142242207)

[Task №3. Loading data into the company table 3](#_Toc142242208)

[Task №4. Company classification 4](#_Toc142242209)

[Task №5. Finding quarterly sales amount by company, and product category 5](#_Toc142242210)

[Task №6. Initial data preparation 6](#_Toc142242211)

[Task №7. Changing plan data 8](#_Toc142242212)

[Task №8. Plan data approval 10](#_Toc142242213)

[Task №9. Data preparation for plan-fact analysis in Q1 2014 12](#_Toc142242214)

[Query for data loading into company\_sales 12](#_Toc142242215)

[Creating the materialized view 13](#_Toc142242216)

## Task №1. A query/queries to put the data into the ‘country\_managers’ table

**create role** planadmin;

**create role** planmanager;

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

**grant update, insert, delete on** plan\_data **to** planadmin, planmanager;

**grant update, insert, delete on** plan\_status **to** planadmin;

**grant update, insert, delete on** country\_managers **to** planadmin;

**grant update on** plan\_status **to** planmanager;

**grant update on** v\_plan\_edit **to** planmanager;

**create user** ivan **with password** 'sql';

**grant** planadmin **to** ivan;

**create user** sophie **with password** 'sql';

**create user** kirill **with password** 'sql';

**grant** planmanager **to** sophie, kirill;

**insert** **into** 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 **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** ps **using**(productsubcategoryid)

**join** productcategory **as** pc **using**(productcategoryid);

--

**create** **materialized** **view** country2 **as**

**select** **distinct** a.countryregioncode **as** countrycode

**from** address **as** a

**join** customeraddress **as** ca **using** (addressid)

**where** ca.addresstype = 'Main Office';

--

**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 **as** c

**join** customeraddress **as** ca **using** (customerid)

**join** address **as** a **using** (addressid)

**where** ca.addresstype = 'Main Office';

## Task №4. Company classification

**insert** **into** company\_abc

**with**

plan\_year **as** (

**values** (2014) -- planning year

),

sales\_data **as** (

**select** company.id **as** cid,

**extract**(**year** **from** soh.orderdate) **as** **year**,

**sum**(soh.subtotal) **as** st\_i

**from** salesorderheader **as** soh

**join** customer **using** (customerid)

**join** company **on** customer.companyname = company.cname

**where** **extract**(**year** **from** soh.orderdate) **in**

((**table** plan\_year) - 2, (**table** plan\_year) - 1)

-- date filtering like below should work faster on big amount of data

-- where soh.orderdate between make\_date((table plan\_year) - 2, 1, 1)::date

-- and make\_date((table plan\_year) - 1, 12, 31)::date

**group** **by** 1, 2

),

sales\_totals **as** (

**select** cid,

**year**,

st\_i **as** salestotal,

**sum**(st\_i) **over** (**partition** **by** **year**) **as** s,

**sum**(st\_i) **over** (**partition** **by** **year** **order** **by** st\_i **desc**) **as** srt\_i

**from** sales\_data

)

**select** cid,

salestotal,

**case**

**when** srt\_i <= s \* 0.80 **then** 'A'

**when** srt\_i <= s \* 0.95 **then** 'B'

**else** 'C'

**end** **as** cls,

**year**

**from** sales\_totals

**order** **by** cid, **year**;

A screenshot of a data

Description automatically generated

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

**insert** **into** company\_sales

**with**

plan\_year **as** (

**values** (2014) -- planning year

),

sales\_data **as** (

**select** cmp.id **as** cid,

**extract**(**year** **from** soh.orderdate) **as** **year**,

**extract**(**quarter** **from** soh.orderdate) **as** quarter\_yr,

p2.pcid **as** categoryid,

**sum**(sod.linetotal) **as** salesamt

**from** salesorderheader **as** soh

**join** salesorderdetail **as** sod **using** (salesorderid)

**join** product2 **as** p2 **using** (productid)

**join** customer **as** cst **using** (customerid)

**join** company **as** cmp **on** cst.companyname = cmp.cname

**where** **extract**(**year** **from** soh.orderdate) **in**

((**table** plan\_year) - 2, (**table** plan\_year) - 1)

**group** **by** 1, 2, 3, 4

)

**select** abc.cid,

sd.salesamt,

sd.**year**,

sd.quarter\_yr,

**concat**(sd.**year**, '.', sd.quarter\_yr) **as** qr,

sd.categoryid,

abc.cls **as** ccls

**from** sales\_data **as** sd

**join** company\_abc **as** abc **using** (cid, **year**)

## Task №6. Initial data preparation

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

    # Supportive queries

    qry\_plan\_data\_cleanup = """

        delete from plan\_data

        where quarterid = %s;

        """

    qry\_plan\_status\_cleanup = """

        delete from plan\_status

        where quarterid = %s;

        """

    qry\_plan\_status\_fillin = """

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

        select %s, 'R', countrycode

        from country2 c;

        """

    qry\_plan\_data\_fillin\_N = """

        insert into plan\_data

        with

        quarterly\_sales as (

            select distinct c.countrycode,

                cs.categoryid as pcid,

                cs.qr,

                sum(cs.salesamt) over (partition by c.countrycode, cs.categoryid, cs.qr) as salesamt

            from company\_sales as cs

                join company as c on c.id = cs.cid

            where cs.ccls in ('A', 'B') and cs.year in (%s - 1, %s - 2) and cs.quarter\_yr = %s

        ),

        average\_sales as (

            select distinct countrycode,

                pcid,

                avg(salesamt) over (partition by countrycode, pcid) as salesamt

            from quarterly\_sales

        ),

        plan\_template as (

            select distinct c2.countrycode,

                p2.pcid

            from product2 as p2

                cross join country2 as c2

        )

        select 'N' as versionid,

            pt.countrycode as country,

            %s as quarterid,

            pt.pcid,

            coalesce (avs.salesamt, 0) as salesamt

        from plan\_template as pt

        left join average\_sales as avs using(countrycode, pcid);

        """

    qry\_plan\_data\_fillin\_P = """

        insert into plan\_data

        select 'P' as versionid,

            country,

            quarterid,

            pcid,

            salesamt

        from plan\_data

        where versionid = 'N' and quarterid = %s;

        """

    qry\_plan\_status\_update\_author = """

        update plan\_status

        set author = current\_user

        where quarterid = %s;

        """

    # Setting the connection

    con = psycopg2.connect(database='2023\_plan\_Aliaksandr', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    # Executing queries

    qr = f'{year}.{quarter}'

    try:

        cur.execute(qry\_plan\_data\_cleanup, (qr,))

        cur.execute(qry\_plan\_status\_cleanup, (qr,))

        cur.execute(qry\_plan\_status\_fillin, (qr,))

        cur.execute(qry\_plan\_data\_fillin\_N, (year, year, quarter, qr))

        cur.execute(qry\_plan\_data\_fillin\_P, (qr,))

        cur.execute(qry\_plan\_status\_update\_author, (qr,))

        con.commit()

    except Exception as err:

        con.rollback()

        print(err)

    con.close()

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

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Task №7. Changing plan data

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

    # Supportive queries

    qry\_plan\_status\_lock = """

        update plan\_status as ps

        set status = %s,

            modifieddatetime = current\_timestamp,

            author = current\_user

        from country\_managers as cm

        where ps.country = cm.country

        and status = %s

        and ps.quarterid = %s

        and cm.username = current\_user;

        """

    # Setting the connection

    con = psycopg2.connect(database='2023\_plan\_Aliaksandr', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    # Executing queries

    qr = f'{year}.{quarter}'

    try:

        cur.execute(qry\_plan\_status\_lock, ('L', 'R', qr))

        con.commit()

    except Exception as err:

        con.rollback()

        print(err)

    con.close()

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

    # Supportive queries

    qry\_plan\_status\_unlock = """

        update plan\_status as ps

        set status = %s,

            modifieddatetime = current\_timestamp,

            author = current\_user

        from country\_managers as cm

        where ps.country = cm.country

        and status = %s

        and ps.quarterid = %s

        and cm.username = current\_user

        and ps.author = current\_user;

        """

    # Setting the connection

    con = psycopg2.connect(database='2023\_plan\_Aliaksandr', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    # Executing queries

    qr = f'{year}.{quarter}'

    try:

        cur.execute(qry\_plan\_status\_unlock, ('R', 'L', qr))

        con.commit()

    except Exception as err:

        con.rollback()

        print(err)

    con.close()

A screenshot of a computer

Description automatically generated

## Task №8. Plan data approval

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

    # Supportive queries

    qry\_plan\_data\_cleanup\_A = """

        delete

        from plan\_data as pd

            using country\_managers as cm

        where pd.country = cm.country

            and pd.quarterid = %s

            and pd.versionid = 'A'

            and cm.username = current\_user;

        """

    qry\_plan\_data\_fillin\_A = """

        insert into plan\_data

        select 'A' as versionid,

            country,

            quarterid,

            pcid,

            salesamt

        from plan\_data as pd

        join plan\_status as ps using (quarterid, country)

        join country\_managers as cm using (country)

        where pd.quarterid = %s and pd.versionid = 'P'

        and status = 'R' and cm.username = current\_user;

        """

    qry\_plan\_status\_update\_A = """

        update plan\_status as ps

        set status = %s,

            modifieddatetime = current\_timestamp,

            author = current\_user

        from country\_managers as cm

        where ps.country = cm.country

        and status = %s

        and ps.quarterid = %s

        and cm.username = current\_user

        and ps.author = current\_user;

        """

    # Setting the connection

    con = psycopg2.connect(database='2023\_plan\_Aliaksandr', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    # Executing queries

    qr = f'{year}.{quarter}'

    try:

        cur.execute(qry\_plan\_data\_cleanup\_A, (qr,))

        cur.execute(qry\_plan\_data\_fillin\_A, (qr,))

        cur.execute(qry\_plan\_status\_update\_A, ('A', 'R', qr))

        con.commit()

    except Exception as err:

        con.rollback()

        print(err)

    con.close()

accept\_plan(2014, 1, 'kirill', 'sql')

accept\_plan(2014, 1, 'sophie', 'sql')

A screenshot of a computer screen

Description automatically generated

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

Option chosen:

1. Load data of 2014 into the company\_sales table and include this table in the view.

### Query for data loading into company\_sales

**insert** **into** company\_sales

**with**

period\_y **as** (**values** (2014)), -- plan/fact year

period\_q **as** (**values** (1)), -- plan/fact quarter

sales\_data **as** (

**select** cmp.id **as** cid,

**extract**(**year** **from** soh.orderdate) **as** **year**,

**extract**(**quarter** **from** soh.orderdate) **as** quarter\_yr,

p2.pcid **as** categoryid,

**sum**(sod.linetotal) **as** salesamt

**from** salesorderheader **as** soh

**join** salesorderdetail **as** sod **using** (salesorderid)

**join** product2 **as** p2 **using** (productid)

**join** customer **as** cst **using** (customerid)

**join** company **as** cmp **on** cst.companyname = cmp.cname

**where** **extract**(**year** **from** soh.orderdate) = (**table** period\_y)

**and** **extract**(**quarter** **from** soh.orderdate) = (**table** period\_q)

**group** **by** 1, 2, 3, 4

)

**select** abc.cid,

sd.salesamt,

sd.**year**,

sd.quarter\_yr,

**concat**(sd.**year**, '.', sd.quarter\_yr) **as** qr,

sd.categoryid,

abc.cls **as** ccls

**from** sales\_data **as** sd

**join** company\_abc **as** abc **on** abc.cid = sd.cid **and** abc.**year** = sd.**year** - 1

**where** abc.cls **in** ('A', 'B');

### Creating the materialized view

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

**with**

period\_y **as** (**values** (2014)), -- setting plan/fact year

period\_q **as** (**values** (1)), -- setting plan/fact quarter

sales\_comparison **as** (

**select** cs.qr,

c.countrycode,

cs.categoryid,

pd.salesamt **as** plan,

**sum**(cs.salesamt) **as** fact,

**round**(pd.salesamt - **sum**(cs.salesamt), 0) **as** dev,

**to\_char**((pd.salesamt - **sum**(cs.salesamt))/pd.salesamt \* 100, '990D0%') **as** dev\_p

**from** company\_sales **as** cs

**join** company **as** c **on** c.id = cs.cid

**join** plan\_data **as** pd

**on** pd.country = c.countrycode

**and** pd.pcid = cs.categoryid

**and** pd.quarterid = cs.qr

**where** pd.versionid = 'A'

**group** **by** 1, 2, 3, 4

),

plan\_fact\_template **as** (

**select** **distinct** **concat**((**table** period\_y), '.', (**table** period\_q)) **as** qr,

c2.countrycode,

p2.pcid **as** categoryid

**from** product2 **as** p2

**cross** **join** country2 **as** c2

)

**select** qr **as** **quarter**,

pft.countrycode **as** country,

pc.**name** **as** category\_name,

-- plan,

-- fact,

dev,

dev\_p

**from** plan\_fact\_template **as** pft

**join** productcategory **as** pc **on** pc.productcategoryid = pft.categoryid

**left** **join** sales\_comparison **using** (countrycode, categoryid, qr)

**order** **by** pft.countrycode, pc.**name**;

A screenshot of a computer

Description automatically generated