1. **Task №1. Access settings**

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

**grant** **update**, **insert**, **delete** **on** public.plan\_data, public.plan\_status, public.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** **update**, **insert**, **delete** **on** public.plan\_data **to** planmanager;

**grant** **update** **on** public.plan\_status, public.v\_plan\_edit **to** planmanager;

**create** **user** ivan **with** **password** 'sql0';

**create** **user** sophie **with** **password** 'sql1';

**create** **user** kirill **with** **password** 'sql2';

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

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

**insert** **into** public.country\_managers (username, country)

**values** ('sophie','US'), ('sophie', 'CA');

**insert** **into** public.country\_managers (username, country)

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

Text

Description automatically generated

1. **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

**left** **join** productsubcategory **as** psc **using**(productsubcategoryid)

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

**with** **no** **data**;

**refresh** **materialized** **view** product2;

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

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

**with** **no** **data**;

**refresh** **materialized** **view** country2;

**grant** **select** **on** product2, country2 **to** planadmin, planmanager;

Text

Description automatically generated

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

Given that there are 2 companies with the same name and different addresses – “Friendly Bike Shop” and “Sports Products Store” I’ve made an assumption that we can take the first occurrence in order to avoid further duplicates as per <https://www.coursera.org/learn/sql/discussions/weeks/6/threads/qeQgo-pSEeuoUw6CZEjA2w>

**insert** **into** company (cname, countrycode, city)

**select** **distinct** c.companyname **as** cname, a.countryregioncode **as** countrycode, a.city

**from** (**select** **min**(customerid) **as** customerid, companyname **from** customer **group** **by** companyname) **as** c

**join** customeraddress **as** ca **on** c.customerid = ca.customerid

**join** address **as** a **on** ca.addressid = a.addressid

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

Text

Description automatically generated

1. **Task №4. Company classification by annual amount of orders**

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

**select** cid,

s **as** salestotal,

**case**

**when** srt <= sa **then** 'A'

**when** srt <= sb **then** 'B'

**else** 'C'

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

sales\_total.yr **as** **year**

**from**

(**select** **date\_part**('y', s.orderdate) **as** yr, **sum**(s.subtotal) **as** s, **sum**(s.subtotal) \* 0.8 **as** sa, **sum**(s.subtotal) \* 0.95 **as** sb

**from** salesorderheader **as** s

**join** customer **as** c **on** c.customerid = s.customerid

**join** company **as** cm **on** cm.cname = c.companyname

**group** **by** **date\_part**('y', s.orderdate)

**having** **date\_part**('y', s.orderdate) **in** (2012, 2013)) **as** sales\_total

**join**

(**select** cid, cname, yr, st,

**sum**(st) **over** (**partition** **by** yr **order** **by** yr **desc**, st **desc**, row\_num) **as** srt

**from**

(**select** cm.id **as** cid,

cm.cname,

**date\_part**('y', s.orderdate) **as** yr,

**sum**(s.subtotal) **as** st,

**row\_number**() **over**(**order** **by** **date\_part**('y', s.orderdate) **desc**, **sum**(s.subtotal) **desc**) **as** row\_num

**from** salesorderheader **as** s

**join** customer **as** c **on** c.customerid = s.customerid

**join** company **as** cm **on** cm.cname = c.companyname

**group** **by** cm.id, cm.cname, **date\_part**('y', s.orderdate)

**having** **date\_part**('y', s.orderdate) **in** (2012, 2013)

**order** **by** yr **desc**, st **desc**) **as** data\_st) **as** sales\_rating

**on** sales\_total.yr = sales\_rating.yr;

Text

Description automatically generated

Table

Description automatically generated with low confidence

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

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

**select**

cm.id **as** cid,

**sum**(sd.linetotal) **as** salesamt,

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

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

''|| **date\_part**('y', sh.orderdate) || '.' || **date\_part**('quarter', sh.orderdate) **as** qr,

p2.pcid **as** categoryid,

cabc.cls **as** ccls

**from**

salesorderheader **as** sh

**join** salesorderdetail **as** sd **on** sd.salesorderid = sh.salesorderid

**join** customer **as** c **on** c.customerid = sh.customerid

**join** company **as** cm **on** cm.cname = c.companyname

**join** product2 **as** p2 **on** p2.productid = sd.productid

**join** company\_abc **as** cabc **on** cabc.cid = cm.id **and** cabc."year" = **date\_part**('y', sh.orderdate)

**group** **by** **date\_part**('y', sh.orderdate),

**date\_part**('quarter', sh.orderdate),

''|| **date\_part**('y', sh.orderdate) || '.' || **date\_part**('quarter', sh.orderdate),

cm.id, p2.pcid, cabc.cls;

Text, letter

Description automatically generated

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

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

    con = psycopg2.connect(database='2020\_plans\_Yaroslav', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    # delete plan data related to the target year and quarter

    query = f'''

            delete from plan\_data as pd where pd.quarterid = '{year}.{quarter}'

            '''

    cur.execute(query)

    # delete records in plan status related to the target quarter

    query = f'''

            delete from plan\_status as ps where right(ps.quarterid, 1) = '{quarter}'

            '''

    cur.execute(query)

    # create planning status records

    query = f'''

            insert into plan\_status

            select distinct

                '{year}.{quarter}' as quarterid,

                'R' as status,

                current\_timestamp as modifieddatetime,

                current\_user as author,

                c.countrycode as country

            from company as c

            '''

    cur.execute(query)

    # generate plan data

    query = f'''

            insert into plan\_data

            select

                'N' as versionid,

                coalesce(data\_avg.countrycode, country\_category.countrycode) as country,

                '{year}.{quarter}' as quarterid,

                coalesce(data\_avg.categoryid, country\_category.categoryid) as pcid,

                coalesce(data\_avg.avg, 0) as salesamt

            from

                (select

                    data.countrycode, data.categoryid,

                    avg(s)

                from

                    (select

                        cs."year" as yr,

                        cs.quarter\_yr as quarter\_yr,

                        c.countrycode as countrycode,

                        cs.categoryid as categoryid,

                        sum(salesamt) as s

                    from company\_sales as cs

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

                        where cs."year" in ({year}-1, {year}-2) and quarter\_yr = {quarter} and cs.ccls != 'C'

                    group by cs."year", cs.quarter\_yr, c.countrycode, cs.categoryid) as data

                group by data.countrycode, data.categoryid

                order by data.countrycode) as data\_avg

            right join

                (select c.countryregioncode as countrycode, p.productcategoryid as categoryid

                from country2 as c

                cross join productcategory as p) as country\_category

            on country\_category.countrycode = data\_avg.countrycode and country\_category.categoryid = data\_avg.categoryid

            '''

    cur.execute(query)

    # copy data from version N into version P

    query = f'''

            insert into plan\_data

            select 'P' as versionid,

                    plan\_data.country,

                    plan\_data.quarterid,

                    plan\_data.pcid,

                    plan\_data.salesamt

            from plan\_data where plan\_data.versionid = 'N'

            '''

    cur.execute(query)

    con.commit()

    con.close()

    print('Done.')

start\_planning(2014, 1, 'ivan', 'sql0')

Table

Description automatically generated with medium confidence

Table

Description automatically generated

Table

Description automatically generated

1. **Task №7. Changing plan data**

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

    con = psycopg2.connect(database='2020\_plans\_Yaroslav', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    query = f'''

            update plan\_status

            set status = 'L',

                modifieddatetime = current\_timestamp,

                author = current\_user

                where quarterid = '{year}.{quarter}' and country in (select cm.country from country\_managers as cm where cm.username = current\_user)

        '''

    cur.execute(query)

    con.commit()

    con.close()

    print('Done.')

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

    con = psycopg2.connect(database='2020\_plans\_Yaroslav', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    query = f'''

            update plan\_status

            set status = 'R',

                modifieddatetime = current\_timestamp,

                author = current\_user

                where quarterid = '{year}.{quarter}' and country in (select cm.country from country\_managers as cm where cm.username = current\_user)

        '''

    cur.execute(query)

    con.commit()

    con.close()

    print('Done.')

set\_lock(2014, 1, 'sophie', 'sql1')

set\_lock(2014, 1, 'kirill', 'sql2')

Table

Description automatically generated

Increased planned sales by 30% when logged as kirill (before saving):

Table

Description automatically generated

remove\_lock(2014, 1, 'sophie', 'sql1')

remove\_lock(2014, 1, 'kirill', 'sql2')

1. **Task №8. Plan data approval**

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

    con = psycopg2.connect(database='2020\_plans\_Yaroslav', user=user, password=pwd, host='localhost')

    cur = con.cursor()

    # delete plan data with version A

    query = f'''

            delete from plan\_data as pd

            where

                right(pd.quarterid, 1) = '{quarter}' and

                pd.versionid = 'A' and

                pd.country in (select cm.country from country\_managers as cm where cm.username = current\_user)

            '''

    cur.execute(query)

    # save a copy into plan data

    query = f'''

            insert into plan\_data

            select 'A' as versionid,

                    pd.country,

                    pd.quarterid,

                    pd.pcid,

                    pd.salesamt

            from plan\_data as pd

            where

                pd.quarterid = '{year}.{quarter}' 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.status = 'R' and ps.quarterid = '{year}.{quarter}')

            '''

    cur.execute(query)

    # update plan status

    query = f'''

            update plan\_status

            set status = 'A',

                modifieddatetime = current\_timestamp,

                author = current\_user

                where quarterid = '{year}.{quarter}' and country in (select cm.country from country\_managers as cm where cm.username = current\_user)

        '''

    cur.execute(query)

    con.commit()

    con.close()

    print('Done.')

accept\_plan(2014, 1, 'sophie', 'sql1')

accept\_plan(2014, 1, 'kirill', 'sql2')

v\_plan view screenshot under sophie:

Table

Description automatically generated

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

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

**select**

quarterid **as** **quarter**,

sales\_plan.country,

category\_name **as** category\_name,

**round**(sales\_plan.salesamt - sales\_fact.salesamt, 0) **as** dev,

**case**

**when** sales\_plan.salesamt = 0 **then** **null**

**else** ''|| **round**((sales\_plan.salesamt - sales\_fact.salesamt)/sales\_plan.salesamt \* 100, 0) || '%'

**end** **as** dev\_perc

**from**

(**select**

cm.countrycode **as** country,

p2.pcid,

p2.pcname **as** category\_name,

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

**from**

salesorderheader **as** sh

**join** salesorderdetail **as** sd **on** sd.salesorderid = sh.salesorderid

**join** customer **as** c **on** c.customerid = sh.customerid

**join** company **as** cm **on** cm.cname = c.companyname

**join** product2 **as** p2 **on** p2.productid = sd.productid

**where**

**date\_part**('y', sh.orderdate) = 2014 **and**

**date\_part**('quarter', sh.orderdate) = 1 **and**

cm.id **in** (**select** c\_abc.cid **from** company\_abc **as** c\_abc **where** c\_abc.cls **in** ('A', 'B') **and** c\_abc."year" = 2013)

**group** **by** cm.countrycode, p2.pcid, p2.pcname) **as** sales\_fact

**join**

(**select** \* **from** plan\_data **as** pd **where** pd.versionid = 'A') **as** sales\_plan

**on** sales\_plan.country = sales\_fact.country **and** sales\_plan.pcid = sales\_fact.pcid

**order** **by** sales\_plan.country

**with no data**;

**refresh** **materialized** **view** mv\_plan\_fact\_2014\_q1;

Text

Description automatically generated

Table

Description automatically generated