## 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)
```

plan_status [postgres@localhost]

18 rows | DDL

WHERE | ORDER BY

| | quarterid | status | modifieddatetime | author | country |
|---|---|---|---|---|---|
| 1 | 2014.2 | R | 2022-08-17 18:15:40.008884 | ivan | AU |
| 2 | 2014.2 | R | 2022-08-17 18:15:40.008884 | ivan | CA |
| 3 | 2014.2 | R | 2022-08-17 18:15:40.008884 | ivan | DE |
| 4 | 2014.2 | R | 2022-08-17 18:15:40.008884 | ivan | FR |
| 5 | 2014.2 | R | 2022-08-17 18:15:40.008884 | ivan | GB |
| 6 | 2014.2 | R | 2022-08-17 18:15:40.008884 | ivan | US |
| 7 | 2014.3 | R | 2022-08-17 19:51:13.628707 | ivan | AU |
| 8 | 2014.3 | R | 2022-08-17 19:51:13.628707 | ivan | CA |
| 9 | 2014.3 | R | 2022-08-17 19:51:13.628707 | ivan | DE |
| 10 | 2014.3 | R | 2022-08-17 19:51:13.628707 | ivan | FR |
| 11 | 2014.3 | R | 2022-08-17 19:51:13.628707 | ivan | GB |
| 12 | 2014.3 | R | 2022-08-17 19:51:13.628707 | ivan | US |
| 13 | 2014.1 | A | 2022-08-17 21:14:56.310019 | kirill | FR |
| 14 | 2014.1 | A | 2022-08-17 21:14:56.310019 | kirill | GB |
| 15 | 2014.1 | A | 2022-08-17 21:14:56.310019 | kirill | DE |
| 16 | 2014.1 | A | 2022-08-17 21:14:56.310019 | kirill | AU |
| 17 | 2014.1 | A | 2022-08-17 21:14:56.327731 | sophie | US |
| 18 | 2014.1 | A | 2022-08-17 21:14:56.327731 | sophie | CA |

plan_data [postgres@localhost]

136 rows | Tx: Auto | DDL

WHERE | ORDER BY

| | versionid | country | quarterid | pcid | salesamt |
|---|---|---|---|---|---|
| 1 | N | AU | 2014.1 | 1 | 51697.27 |
| 2 | N | CA | 2014.1 | 1 | 448609.27 |
| 3 | N | DE | 2014.1 | 1 | 40664.88 |
| 4 | N | FR | 2014.1 | 1 | 69361.23 |
| 5 | N | GB | 2014.1 | 1 | 85558.36 |
| 6 | N | US | 2014.1 | 1 | 1204916.66 |
| 7 | N | AU | 2014.1 | 2 | 157846.44 |
| 8 | N | CA | 2014.1 | 2 | 102625.52 |
| 9 | N | DE | 2014.1 | 2 | 16132.19 |
| 10 | N | FR | 2014.1 | 2 | 34156.94 |
| 11 | N | GB | 2014.1 | 2 | 4735.46 |
| 12 | N | US | 2014.1 | 2 | 1958022.18 |
| 13 | N | AU | 2014.1 | 4 | 429.34 |
| 14 | N | CA | 2014.1 | 4 | 991.57 |
| 15 | N | DE | 2014.1 | 4 | 196.81 |
| 16 | N | FR | 2014.1 | 4 | 232.53 |
| 17 | N | GB | 2014.1 | 4 | 33.77 |
| 18 | N | US | 2014.1 | 4 | 7241.03 |
| 19 | P | AU | 2014.1 | 1 | 51697.27 |
| 20 | N | AU | 2014.2 | 1 | 26952.33 |
| 21 | N | CA | 2014.2 | 1 | 151196.76 |
| 22 | N | FR | 2014.2 | 1 | 37610.51 |
| 23 | N | GB | 2014.2 | 1 | 23266.52 |
| 24 | N | US | 2014.2 | 1 | 1239533.23 |
| 25 | N | AU | 2014.2 | 2 | 147054.02 |
| 26 | N | CA | 2014.2 | 2 | 134514.94 |
| 27 | N | DE | 2014.2 | 2 | 45200.25 |
| 28 | N | FR | 2014.2 | 2 | 94667.68 |
| 29 | N | GB | 2014.2 | 2 | 3288.56 |
| 30 | N | US | 2014.2 | 2 | 1990711.37 |
| 31 | N | AU | 2014.2 | 3 | 12725.37 |
| 32 | N | US | 2014.2 | 3 | 1023036.62 |
| 33 | N | AU | 2014.2 | 4 | 848.42 |
| 34 | N | CA | 2014.2 | 4 | 1040.03 |
| 35 | N | DE | 2014.2 | 4 | 183.93 |
| 36 | N | FR | 2014.2 | 4 | 830.83 |
| 37 | N | GB | 2014.2 | 4 | 62.81 |
| 38 | N | US | 2014.2 | 4 | 16446.16 |
| 39 | P | AU | 2014.2 | 1 | 26952.33 |
| 40 | P | CA | 2014.2 | 1 | 151196.76 |
| 41 | P | FR | 2014.2 | 1 | 37610.51 |
| 42 | P | GB | 2014.2 | 1 | 23266.52 |
| 43 | P | US | 2014.2 | 1 | 1239533.23 |
| 44 | P | AU | 2014.2 | 2 | 147054.02 |
| 45 | P | CA | 2014.2 | 2 | 134514.94 |
| 46 | P | DE | 2014.2 | 2 | 45200.25 |
| 47 | P | FR | 2014.2 | 2 | 94667.68 |
| 48 | P | GB | 2014.2 | 2 | 3288.56 |
| 49 | P | US | 2014.2 | 2 | 1990711.37 |
| 50 | P | AU | 2014.2 | 3 | 12725.37 |

## 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)
```

v_plan_edit [kirill@localhost]

12 rows  Tx: Auto  DDL

WHERE  ORDER BY

| | country | quarterid | pcid | salesamt | versionid |
|---|---|---|---|---|---|
| 1 | AU | 2014.1 | 1 | 51697.27 | P |
| 2 | DE | 2014.1 | 1 | 40664.88 | P |
| 3 | FR | 2014.1 | 1 | 69361.23 | P |
| 4 | GB | 2014.1 | 1 | 85558.36 | P |
| 5 | AU | 2014.1 | 2 | 157846.44 | P |
| 6 | DE | 2014.1 | 2 | 16132.19 | P |
| 7 | FR | 2014.1 | 2 | 34156.94 | P |
| 8 | GB | 2014.1 | 2 | 4735.46 | P |
| 9 | AU | 2014.1 | 4 | 429.34 | P |
| 10 | DE | 2014.1 | 4 | 196.81 | P |
| 11 | FR | 2014.1 | 4 | 232.53 | P |
| 12 | GB | 2014.1 | 4 | 33.77 | P |

## 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)
```

v_plan [sophie@localhost]

6 rows | Tx: Auto | DDL

WHERE | ORDER BY

| | country | pcid | quarterid | salesamt |
|---|---|---|---|---|
| 1 | CA | 1 | 2014.1 | 448609.27 |
| 2 | US | 1 | 2014.1 | 1204916.66 |
| 3 | CA | 2 | 2014.1 | 102625.52 |
| 4 | US | 2 | 2014.1 | 1958022.18 |
| 5 | CA | 4 | 2014.1 | 991.57 |
| 6 | US | 4 | 2014.1 | 7241.03 |

## 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
```

```sql
    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
```

mv_plan_fact_2014_q1 [postgres@localhost]

16 rows | Tx: Auto | DDL | CSV

WHERE | ORDER BY

| | quater | country | categoryname | dev | dev_perc |
|---|---|---|---|---|---|
| 1 | 2014.1 | AU | Bikes | 27502.49 | 0.53199114769503302592 |
| 2 | 2014.1 | AU | Components | 0 | 0 |
| 3 | 2014.1 | AU | Accessories | 0 | 0 |
| 4 | 2014.1 | CA | Bikes | 294018.74 | 0.65540050030620187586 |
| 5 | 2014.1 | CA | Components | -39750.85 | -0.38733884125507963321 |
| 6 | 2014.1 | CA | Accessories | 0 | 0 |
| 7 | 2014.1 | DE | Components | -15293.26 | -0.94799652124107142304 |
| 8 | 2014.1 | DE | Accessories | 0 | 0 |
| 9 | 2014.1 | FR | Bikes | 0 | 0 |
| 10 | 2014.1 | FR | Components | -30765.39 | -0.90070685488805496043 |
| 11 | 2014.1 | FR | Accessories | 0 | 0 |
| 12 | 2014.1 | GB | Bikes | 48616.25 | 0.56822325720128342806 |
| 13 | 2014.1 | GB | Accessories | 0 | 0 |
| 14 | 2014.1 | US | Bikes | 73274.59 | 0.06081299431945774573 |
| 15 | 2014.1 | US | Components | -76637.24 | -0.03914012863735792819 |
| 16 | 2014.1 | US | Accessories | 0 | 0 |