

1. Task No 1. Access settings

```
create user ivan with password 'ivan1';
create user sophie with password 'sophie1';
create user kirill with password 'kirill1';

grant planadmin to ivan;
grant planmanager to sophie;
grant planmanager to kirill;

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

grant select, update, insert, delete on plan_data to planadmin;
grant select, update, insert, delete on plan_data to planmanager;

grant select, update, insert, delete on plan_status to planadmin;
grant select, update on plan_status to planmanager;

grant select, update, insert, delete on country_managers to planadmin;
grant select on country_managers to planmanager;

grant select, update on v_plan_edit to planmanager;

grant select on v_plan to planmanager;

insert into country_managers (username, country) values ('sophie', 'US');
insert into country_managers (username, country) values ('sophie', 'CA');

insert into country_managers (username, country) values ('kirill', 'FR');
insert into country_managers (username, country) values ('kirill', 'GB');
insert into country_managers (username, country) values ('kirill', 'DE');
insert into country_managers (username, country) values ('kirill', 'AU');
```

	username	country
1	sophie	US
2	sophie	CA
3	kirill	FR
4	kirill	GB
5	kirill	DE
6	kirill	AU

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

```

--task 2--
create materialized view product2 as
select p3.productcategoryid as pcid, p1.productid as productcid,
p3.name as pcname, p1.name as pname
from product p1 join productsubcategory p2 using (productsubcategoryid)
join productcategory p3 using(productcategoryid);

create materialized view country2 as
select distinct ad.countryregioncode
from customer as c
join customeraddress ca using(customerid)
join address ad using(addressid)
where ca.addresstype = 'Main Office';

--permission to managers and administrators
grant select on product2 to planadmin;
grant select on country2 to planadmin;
grant select on product2 to planmanager;
grant select on country2 to planmanager;

```

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

```

--task 3--
insert into company (cname, countrycode, city)
select distinct c.companyname as cname, ad.countryregioncode, ad.city
from customer c join customeraddress ca using(customerid)
join address ad using(addressid)
where ca.addresstype = 'Main Office';

```

### 4. Task №4. Company classification

```

insert into company_abc (cid, salestotal, cls, year)
select cid, sale_company as salestotal, case when srt <= sa then 'A' when srt <= sb then 'B' else 'C' end as cls,
year
from (select *, sum(ratings.sale_company)over(partition by year order by year desc rows between unbounded preceding and current row) as SRT
from
(select c.id as cid, sum(sales.subtotal) as sale_company,extract(year from sales.orderdate) as year
from salesorderheader sales
join customer cus using(customerid)
join company c on cus.companyname = c.cname
where extract(year from sales.orderdate) = 2012 or extract (year from sales.orderdate) = 2013
group by c.id, extract(year from sales.orderdate)
order by year desc, sale_company desc
)ratings
join
(select sum(salestotal.subtotal) as totalsales, extract(year from salestotal.orderdate) as year,
sum(salestotal.subtotal)*0.8 as sa, sum(salestotal.subtotal)*0.95 sb
from salesorderheader salestotal
join customer cus using (customerid)
join company c on cus.companyname = c.cname
where extract(year from salestotal.orderdate) = 2012 or extract(year from salestotal.orderdate) = 2013
group by extract(year from salestotal.orderdate)
order by extract(year from salestotal.orderdate) desc) total using (year))total order by cid;

```

	cid	salestotal	cls	year
1	1	249.54	C	2,013
2	2	190,732.7335	A	2,012
3	2	68,243.9474	B	2,013
4	3	34,995.753	B	2,012
5	3	3,975.4005	C	2,013
6	4	42,309.3031	B	2,013
7	6	42,746.2971	B	2,013
8	7	113,497.0245	A	2,013
9	8	14,021.208	C	2,013
10	9	6,591.8124	C	2,012
11	9	98.886	C	2,013
12	10	2,854.0183	C	2,012
13	10	21,911.193	C	2,013
14	11	11,295.7189	C	2,012
15	12	3,288.918	C	2,013
16	13	256,319.6209	A	2,013
17	13	278,173.4175	A	2,012
18	15	106,723.1923	A	2,013
19	16	25,106.9619	B	2,013
20	17	8,429.364	C	2,013
21	19	10,655.43	C	2,013

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

```
insert into company_sales (cid, salesamt, year, quarter_yr, qr, categoryid, ccls)
select c.id as tid, sum(sd.linnetotal) as salesamount, extract(year from sh.orderdate) as year,
extract(quarter from sh.orderdate) as quarter_yr,
to_char(sh.orderdate ::date, 'YYYY.Q') as qr, p2.pcid as categoryid, c_abc.cls
from salesorderdetail sd
join salesorderheader sh using(salesorderid)
join customer cu using (customerid)
join company c on cu.companyname = c.cname
join product2 p2 using(productid)
join company_abc c_abc on c_abc.cid = c.id and c_abc.year = extract(year from sh.orderdate)
where extract(year from sh.orderdate) = 2012 or extract(year from sh.orderdate) = 2013
group by c.id, extract(year from sh.orderdate), extract(quarter from sh.orderdate),
qr, categoryid, c_abc.cls;
```

## 6. Task №6. Initial data preparation

```
#start planning function
def start_planning(year, quarter, user, pwd):
    #connect to database
    con = psycopg2.connect(database = "joyce_planning_2020", user = user, password = pwd, host = "localhost")
    cur = con.cursor()

    #step1: Delete plan data from the data_table related to target year and quarter
    cur.execute("delete from plan_data where quarterid=cast('%s.%s as varchar(6))", (year, quarter,))
    con.commit()

    #step2: Delete plan_status related to the target quarter
    cur.execute("delete from plan_status where quarterid=cast('%s.%s as varchar(6))", (year,quarter,))
    con.commit()

    #step3: create planning_status records for the selected quarter
    cur.execute("""insert into plan_status (quarterid, country, status)
select distinct %s||'. '||%s quarterid, c2.countryregioncode country, 'R' from country2 c2
left join (select cs.qr, c.countrycode
from company_sales cs join company c on c.id=cs.cid where quarter_yr=%s)t
on t.countrycode = c2.countryregioncode""", (year,quarter,quarter,))

    con.commit()

    # step4: generate version n of planning data
    cur.execute("""insert into plan_data
select 'N' versionid, countryregioncode as country, %s||'. '||%s, productcategoryid as pcid , coalesce(salesamt ,0) as salesamt
from country2 cross join productcategory p left join
(select c.countrycode country, categoryid pcid, sum(salesamt)/2 as salesamt
from company_sales cs left join company as c on cs.cid = c.id where quarter_yr = %s and ccls != 'C'
group by c.countrycode , categoryid
order by c.countrycode , categoryid) as t on t.country = countryregioncode and t.pcid = productcategoryid""", (year, quarter, quarter,))
```

```

        order by countryregioncode, productcategoryid""", (year, quarter, quarter))
    con.commit()
    # #step5: copy version n
    cur.execute("""insert into plan_data(versionid, country, quarterid, pcid, salesamt)
                select 'P' versionid, p.country, p.quarterid, p.pcid, p.salesamt from plan_data p""")
    con.commit()

start_planning(2014, 1, 'ivan', 'ivani')

```

	quarterid	status	modifieddatetime	author	country	
1	2014.1	R	2020-12-05 08:34:52	ivan	AU	
2	2014.1	R	2020-12-05 08:34:52	ivan	CA	
3	2014.1	R	2020-12-05 08:34:52	ivan	DE	
4	2014.1	R	2020-12-05 08:34:52	ivan	FR	
5	2014.1	R	2020-12-05 08:34:52	ivan	GB	
6	2014.1	R	2020-12-05 08:34:52	ivan	US	

	versionid	country	quarterid	pcid	salesamt
1	N	AU	2014.1	1	0
2	N	AU	2014.1	2	0
3	N	AU	2014.1	3	0
4	N	AU	2014.1	4	0
5	N	CA	2014.1	1	1,016,439.37
6	N	CA	2014.1	2	130,085.46
7	N	CA	2014.1	3	19,446.01
8	N	CA	2014.1	4	4,066.3
9	N	DE	2014.1	1	0
10	N	DE	2014.1	2	0
11	N	DE	2014.1	3	0
12	N	DE	2014.1	4	0
13	N	FR	2014.1	1	112,661.9
14	N	FR	2014.1	2	28,607.34
15	N	FR	2014.1	3	4,641.05
16	N	FR	2014.1	4	772.01
17	N	GB	2014.1	1	105,766.91
18	N	GB	2014.1	2	19,404.26
19	N	GB	2014.1	3	4,146.22
20	N	GB	2014.1	4	876.03
21	N	US	2014.1	1	4,131,941.37
22	N	US	2014.1	2	438,482.92
23	N	US	2014.1	3	56,479.27
24	N	US	2014.1	4	13,199.87
25	P	AU	2014.1	1	0
26	P	AU	2014.1	2	0
27	P	AU	2014.1	3	0
28	P	AU	2014.1	4	0
29	P	CA	2014.1	1	1,016,439.37
30	P	CA	2014.1	2	130,085.46
31	P	CA	2014.1	3	19,446.01
32	P	CA	2014.1	4	4,066.3
33	P	DE	2014.1	1	0
34	P	DE	2014.1	2	0
35	P	DE	2014.1	3	0
36	P	DE	2014.1	4	0

## 7. Task №7. Changing the plan data

```
#set_lock
def set_lock(year, quarter, user, pwd):
    #connect to database
    con = psycopg2.connect(database = "joyce_planning_2020", user = user, password = pwd, host = "localhost")
    cur = con.cursor()

    #change status from R to L
    cur.execute("""update plan_status
        set status = 'L',
        author = %s,
        modifieddatetime = current_timestamp
        from country_managers
        where quarterid = %s||'.'||%s
        and country_managers.username = %s
        and country_managers.country = plan_status.country""",(user, year, quarter,user,))

    con.commit()
    con.close()

set_lock(2014, 1, 'kirill', 'kirill1')
set_lock(2014, 1, 'sophie', 'sophie1')

#remove_lock
def remove_lock(year, quarter, user, pwd):
    #connect to database
    con = psycopg2.connect(database = "joyce_planning_2020", user = user, password = pwd, host = "localhost")
```

```

cur = con.cursor()

#change status from L to R
cur.execute("""update plan_status
              set status = 'R',
              author = %s,
              modifieddatetime = current_timestamp
              from country_managers
              where quarterid = %s||'.'||%s
              and country_managers.username = %s
              and country_managers.country = plan_status.country""",(user, year, quarter,user,))

con.commit()
con.close()

```

(a) Screenshots before increasing the plan sales by 30%.

plan_status						
	quarterid	status	modifieddatetime	author	country	
1	2014.1	L	2020-12-05 13:12:56	kirill	AU	
2	2014.1	L	2020-12-05 13:12:56	kirill	DE	
3	2014.1	L	2020-12-05 13:12:56	kirill	FR	
4	2014.1	L	2020-12-05 13:12:56	kirill	GB	
5	2014.1	L	2020-12-05 13:12:56	sophie	CA	
6	2014.1	L	2020-12-05 13:12:56	sophie	US	

i. Logged in as Kirill

v_plan_edit						
	country	quarterid	pcid	salesamt	versionid	
1	AU	2014.1	1	0	P	
2	AU	2014.1	2	0	P	
3	AU	2014.1	3	0	P	
4	AU	2014.1	4	0	P	
5	DE	2014.1	1	0	P	
6	DE	2014.1	2	0	P	
7	DE	2014.1	3	0	P	
8	DE	2014.1	4	0	P	
9	FR	2014.1	1	112,661.9	P	
10	FR	2014.1	2	28,607.34	P	
11	FR	2014.1	3	4,641.05	P	
12	FR	2014.1	4	772.01	P	
13	GB	2014.1	1	105,766.91	P	
14	GB	2014.1	2	19,404.26	P	
15	GB	2014.1	3	4,146.22	P	
16	GB	2014.1	4	876.03	P	

ii. Logged in as Sophie

v\_plan\_edit | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC country	ABC quarterid	123 pcid	123 salesamt	ABC versionid
1	CA	2014.1	1	1,016,439.37	P
2	CA	2014.1	2	130,085.46	P
3	CA	2014.1	3	19,446.01	P
4	CA	2014.1	4	4,066.3	P
5	US	2014.1	1	4,131,941.37	P
6	US	2014.1	2	438,482.92	P
7	US	2014.1	3	56,479.27	P
8	US	2014.1	4	13,199.87	P

(b) Screenshots after increasing the plan sales by 30%.

```
--increase sales amount by 30%--
update v_plan_edit
set salesamt = salesamt*1.3;
```

i. Logged in as Kirill

<postgres> final.sql | v\_plan\_edit | plan\_status

Properties | Data | ER Diagram | postgres

v\_plan\_edit | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC country	ABC quarterid	123 pcid	123 salesamt	ABC versionid
1	AU	2014.1	1	0	P
2	AU	2014.1	2	0	P
3	AU	2014.1	3	0	P
4	AU	2014.1	4	0	P
5	DE	2014.1	1	0	P
6	DE	2014.1	2	0	P
7	DE	2014.1	3	0	P
8	DE	2014.1	4	0	P
9	FR	2014.1	1	146,460.47	P
10	FR	2014.1	2	37,189.54	P
11	FR	2014.1	3	6,033.37	P
12	FR	2014.1	4	1,003.61	P
13	GB	2014.1	1	137,496.98	P
14	GB	2014.1	2	25,225.54	P
15	GB	2014.1	3	5,390.09	P
16	GB	2014.1	4	1,138.84	P

ii. Logged in as Sophie

v_plan_edit Enter a SQL expression to filter results (use Ctrl+Space)						
	ABC country	ABC quarterid	123 pcid	123 salesamt	ABC versionid	
1	CA	2014.1	1	1,321,371.18	P	
2	CA	2014.1	2	169,111.1	P	
3	CA	2014.1	3	25,279.81	P	
4	CA	2014.1	4	5,286.19	P	
5	US	2014.1	1	5,371,523.78	P	
6	US	2014.1	2	570,027.8	P	
7	US	2014.1	3	73,423.05	P	
8	US	2014.1	4	17,159.83	P	

(c) Screenshot after running remove\_lock function

plan_status Enter a SQL expression to filter results (use Ctrl+Space)						
	ABC quarterid	ABC status	modifieddatetime	ABC author	ABC country	
1	2014.1	R	2020-12-05 15:07:57	kirill	AU	
2	2014.1	R	2020-12-05 15:07:57	kirill	DE	
3	2014.1	R	2020-12-05 15:07:57	kirill	FR	
4	2014.1	R	2020-12-05 15:07:57	kirill	GB	
5	2014.1	R	2020-12-05 15:07:57	sophie	CA	
6	2014.1	R	2020-12-05 15:07:57	sophie	US	

## 8. Task №8. Plan data approval

```
def accept_plan(year, quarter, user, pwd):
    #connect to database
    con = psycopg2.connect(database = "joyce_planning_2020", user = user, password = pwd, host = "localhost")
    cur = con.cursor()

    #step1: clear A version of plan data for specific quarter and countries accessible to the current user
    cur.execute("""delete from plan_data p
                  using country_managers cm where p.country = cm.country and
                  cm.username = %s and p.versionid = %s||'.'||%s""", (user,year, quarter,))
    con.commit()

    #step2: read data available to the current user from the version P and save its copy as the version A
    cur.execute("""insert into plan_data (versionid, country, quarterid, pcid, salesamt)
                  select 'A', pd.country, pd.quarterid, pd.pcid, pd.salesamt from plan_data pd
                  join country_managers cm using(country)
                  where cm.username = %s and pd.versionid = 'P' and pd.quarterid = %s||'.'||%s """,(user,year, quarter,))
    con.commit()

    #step 3: change data status from R to A
    cur.execute("""update plan_status
                  set status = 'A',
                  author = %s,
                  modifieddatetime = current_timestamp
                  from country_managers
                  where quarterid = %s||'.'||%s
                  and country_managers.username = %s
                  and country_managers.country = plan_status.country""",(user, year, quarter,user,))
    con.commit()
    con.close()

accept_plan(2014, 1, 'kirill', 'kirill!')
accept_plan(2014, 1, 'sophie', 'sophie!')
```



v_plan					
Enter a SQL expression to filter results (use Ctrl+Space)					
	ABC country	123 pcid	ABC quarterid	123 salesamt	
1	CA	1	2014.1	1,321,371.18	
2	CA	2	2014.1	169,111.1	
3	CA	3	2014.1	25,279.81	
4	CA	4	2014.1	5,286.19	
5	US	1	2014.1	5,371,523.78	
6	US	2	2014.1	570,027.8	
7	US	3	2014.1	73,423.05	
8	US	4	2014.1	17,159.83	

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

```
create materialized view mv_plan_fact_2014_q1 as
select t.* from country2 cross join productcategory p3
left join(select plan.quarterid, plan.country, plan.categoryname, plan.salesamt - actual.salesamt as dev,
case when plan.salesamt = 0 then '-infinity' else ((plan.salesamt-actual.salesamt)/plan.salesamt::float)*100 end as dev_perc
from(select pd.quarterid, pd.country country, pc.name categoryname, pd.salesamt salesamt from plan_data pd
join productcategory pc on pc.productcategoryid = pd.pcid
where pd.versionid = 'P'and pd.quarterid = '2014.1')plan
join (select distinct c.countrycode country, p2.pcname pcname, p2.pcid, to_char(sh.orderdate::date, 'YYYY.Q')as quarterid,
sum(sd.linnetotal)over(partition by c.countrycode, p2.pcid, extract(quarter from sh.orderdate)) as salesamt
from salesorderdetail sd
join salesorderheader sh using(salesorderid)
join customer cu using(customerid)
join company c on cu.companyname = c.cname
join product2 p2 using(productid)
join company_abc ca
on ca.cid = c.id and ca.year = 2013
where to_char(sh.orderdate::date, 'YYYY.Q') = '2014.1' and ca.cis <> 'C')actual
on plan.country = actual.country and plan.categoryname = actual.pcname)t
on country2.countryregioncode = t.country and p3."name" = t.categoryname;
```

mv\_plan\_fact\_2014\_q1 Enter a SQL expression to filter results (use Ctrl+Space)

		country	categoryname	dev	dev_perc
1	2014.1	AU	Bikes	-313,362.198	-Infinity
2	2014.1	AU	Components	-27,649.56	-Infinity
3	2014.1	AU	Clothing	-5,364.080976	-Infinity
4	2014.1	AU	Accessories	-2,994.54	-Infinity
5	2014.1	CA	Bikes	372,186.511155	28.1666890264
6	2014.1	CA	Components	-13,469.136592	-7.9646673648
7	2014.1	CA	Clothing	-11,304.088156	-44.7158746684
8	2014.1	CA	Accessories	-11,598.8551	-219.4180515645
9	2014.1	DE	Bikes	-384,604.899408	-Infinity
10	2014.1	DE	Components	-50,266.434	-Infinity
11	2014.1	DE	Clothing	-14,808.721717	-Infinity
12	2014.1	DE	Accessories	-7,742.522656	-Infinity
13	2014.1	FR	Bikes	-432,115.872644	-295.0392502796
14	2014.1	FR	Components	-91,511.492	-246.0678244474
15	2014.1	FR	Clothing	-10,705.625334	-177.4402255124
16	2014.1	FR	Accessories	-8,053.632908	-802.4663871424
17	2014.1	GB	Bikes	-363,033.440532	-264.0301194484
18	2014.1	GB	Components	-53,209.316	-210.934299127
19	2014.1	GB	Clothing	-9,255.793768	-171.7187239545
20	2014.1	GB	Accessories	-6,091.486723	-534.8852097749
21	2014.1	US	Bikes	1,586,310.942219	29.5318611103
22	2014.1	US	Components	-1,323.633088	-0.2322050061
23	2014.1	US	Clothing	-23,284.876091	-31.7133054143
24	2014.1	US	Accessories	-21,572.11704	-125.7128831696