Task №1. Access settings

**revoke** **all** **ON** **all** **tables** **IN** **schema** public **FROM** PUBLIC, planadmin, planmanager;

**drop** **user** **if** **exists** ivan, sophie, kirill;

**truncate** **table** country\_managers;

**grant** **select** **ON** **all** **tables** **IN** **schema** public **TO** planadmin, planmanager;

**grant** **select**, **insert**, **delete**, **update** **ON** plan\_data, plan\_status, country\_managers **TO** planadmin;

**grant** **select**, **insert**, **delete**, **update** **ON** plan\_data **TO** planmanager;

**grant** **select**, **update** **ON** plan\_status, v\_plan\_edit **TO** planmanager;

**grant** **select** **ON** v\_plan **TO** planmanager;

**create** **user** ivan **with** **role** planadmin;

**create** **user** kirill **with** **role** planmanager;

**create** **user** sophie **with** **role** planmanager;

**insert** **into**

country\_managers

**VALUES**

('kirill','FR'),

('kirill','GB'),

('kirill','DE'),

('kirill','AU'),

('sophie', 'US'),

('sophie', 'CA');

Task №2. product2 & country2 materialized views

**drop** **materialized** **view** **IF** **exists** product2;

**drop** **materialized** **view** **IF** **exists** country2;

**create** **materialized** **view** product2 **as**

**select** pcc.categoryid **as** pcid,

p.productid,

pcc.category **as** pcname,

p.**name** **as** pname

**from** product p

**inner** **join** (

**select** psc.productsubcategoryid **as** subcategoryid,

psc.**name** **as** subcategory,

pc.productcategoryid **as** categoryid,

pc.**name** **as** category

**from** productsubcategory psc

**inner** **join** productcategory pc

**on** psc.productcategoryid = pc.productcategoryid

) **as** pcc

**on** p.productsubcategoryid = pcc.subcategoryid;

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

**select** **DISTINCT** a.countryregioncode **as** countrycode

**from** address a

**inner** **join** (

**select** ca.addresstype **as** addresstype,

ca.addressid

**from** customer c

**inner** **join** customeraddress ca

**on** c.customerid = ca.customerid

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

) **as** caa

**on** a.addressid = caa.addressid;

**grant** **select** **ON** product2, country2 **TO** planmanager, planadmin;

Task №3. Loading data into the company table

**drop** **table** **if** **exists** company;

**create** **table** company **as** (

**select** **row\_number**() **over** () **as** id,

cname,

a.countryregioncode **as** countrycode,

a.city

**from** address a

**inner** **join** (

**select** **distinct** c.companyname **as** cname,

ca.addressid

**from** customer c

**inner** **join** customeraddress ca

**on** c.customerid = ca.customerid

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

) **as** caa

**on** a.addressid = caa.addressid

);

Task №4. Company classification

**truncate** **table** company\_abc ;

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

**select** cid,

salestotal,

**case**

**when** **sum**(salestotal) **over** (**partition** **by** "year" **order** **by** salestotal **DESC**) <= 0.8 \* **sum**(salestotal) **over** (**partition** **by** "year")

**then** 'A'

**when** **sum**(salestotal) **over** (**partition** **by** "year" **order** **by** salestotal **DESC**) <= 0.95 \* **sum**(salestotal) **over** (**partition** **by** "year")

**then** 'B'

**else** 'C'

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

**year**

**from** (

**select** c.id **as** cid,

spcy.companyname,

spcy.salestotal,

spcy."year"

**from** company c

**inner** **join** (

**select** **sum**(s.subtotal) **as** salestotal,

**date\_part**('year', s.orderdate) **as** "year",

c.companyname

**from** salesorderheader s

**inner** **join** customer c

**on** s.customerid = c.customerid

**where** c.companyname **notnull** -- ignore not company

**group** **by** c.companyname, "year"

**order** **by** salestotal **DESC**

) **as** spcy

**on** c.cname = spcy.companyname

) **as** s

**where** "year" = 2012 **or** "year" = 2013;

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

**drop** **table** **if** **exists** temp\_4;

**drop** **table** **if** **exists** temp\_3;

**drop** **table** **if** **exists** temp\_2;

**drop** **table** **if** **exists** temp\_1;

**select** customerid,

sod.linetotal,

sod.productid,

**date\_part**('year', soh.orderdate) **as** "year",

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

**date\_part**('year', soh.orderdate) || '.' || **date\_part**('quarter', soh.orderdate) **as** qr

**into** temp\_1

**from** salesorderdetail sod

**inner** **join** salesorderheader soh

**on** sod.salesorderid = soh.salesorderid;

**select** \* **from** temp\_1 **where** "year" = 2012 **or** "year" = 2013;

**select** t1.\*, cc.cid

**into** temp\_2

**from** temp\_1 t1

**inner** **join** (

**select** c2.id **as** cid,

c.customerid

**from** customer c

**inner** **join** company c2

**on** c.companyname = c2.cname

) **as** cc

**on** t1.customerid = cc.customerid;

**select** \* **from** temp\_2 **where** "year" = 2012 **or** "year" = 2013;

**select** t2.\*,

p2.pcid

**into** temp\_3

**from** temp\_2 t2

**inner** **join** product2 p2

**on** t2.productid = p2.productid;

**select** \* **from** temp\_3 **where** "year" = 2012 **or** "year" = 2013;

**select** t3.\*,

ccls.cls

**into** temp\_4

**from** temp\_3 t3

**inner** **join** company\_abc ccls

**on** t3."year" = ccls."year" **and** t3.cid = ccls.cid ;

**select** \* **from** temp\_4; -- no need to specify year, coz we assigned categories only for 2012 and 2013

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

**select** cid,

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

"year",

quarter\_yr,

qr,

pcid **as** categoryid,

cls

**from** temp\_4

**group** **by** qr, cid, pcid, "year", quarter\_yr, cls;

Task №6. Initial data preparation

Listing

def get\_quarter\_id(qr, year):

return str(year) + '.' + str(qr)

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

conn = psycopg2.connect(database='planning', user=user, password=pwd, host='localhost')

cur = conn.cursor()

# clear tables

clear\_plan\_data\_qr = """DELETE FROM plan\_data WHERE quarterid = %s"""

clear\_plan\_status\_qr = """DELETE FROM plan\_status WHERE quarterid = %s"""

current\_quarter\_id = get\_quarter\_id(quarter, year)

cur.execute(clear\_plan\_data\_qr, (current\_quarter\_id,))

conn.commit()

cur.execute(clear\_plan\_status\_qr, (current\_quarter\_id,))

conn.commit()

# fill plan\_status

timestamp = datetime.datetime.now().timestamp()

countries\_qr = """select \* from country2"""

cur.execute(countries\_qr)

plan\_status\_records = []

for record in cur:

plan\_status\_records.append([current\_quarter\_id, timestamp, user, record[0]])

insert\_plan\_status\_qr = """INSERT INTO plan\_status VALUES (%s, 'R', to\_timestamp(%s), %s, %s)"""

for record in plan\_status\_records:

cur.execute(insert\_plan\_status\_qr, record)

conn.commit()

# fill plan\_data

cur.execute('select productcategoryid from productcategory')

categories = [item[0] for item in cur.fetchall()]

cur.execute('select countrycode from country2')

countries = [item[0] for item in cur.fetchall()]

combinations = itertools.product(categories, countries)

planned\_salesamt\_qr = """select categoryid, c.countrycode, sum(salesamt)/2 as salesamt from company\_sales cs

inner join company c

on cs.cid = c.id

where (qr = %s or qr = %s) and ccls in ('A', 'B')

group by categoryid, c.countrycode"""

cur.execute(planned\_salesamt\_qr, [get\_quarter\_id(quarter, year-2), get\_quarter\_id(quarter, year-1)])

average\_saleamn = cur.fetchall()

insert\_plan\_data\_qr = """INSERT INTO plan\_data VALUES (%s, %s, %s, %s, %s)"""

for comb in combinations:

res = list(filter(lambda item: item[0] == comb[0] and item[1] == comb[1], average\_saleamn))

if res:

row\_data = res[0]

cur.execute(insert\_plan\_data\_qr, ['N', row\_data[1], current\_quarter\_id, row\_data[0], row\_data[2]])

cur.execute(insert\_plan\_data\_qr, ['P', row\_data[1], current\_quarter\_id, row\_data[0], row\_data[2]])

else:

cur.execute(insert\_plan\_data\_qr, ['N', comb[1], current\_quarter\_id, comb[0], 0])

cur.execute(insert\_plan\_data\_qr, ['P', comb[1], current\_quarter\_id, comb[0], 0])

conn.commit()

Functions calls

start\_planning(2014, 1, 'kirill')

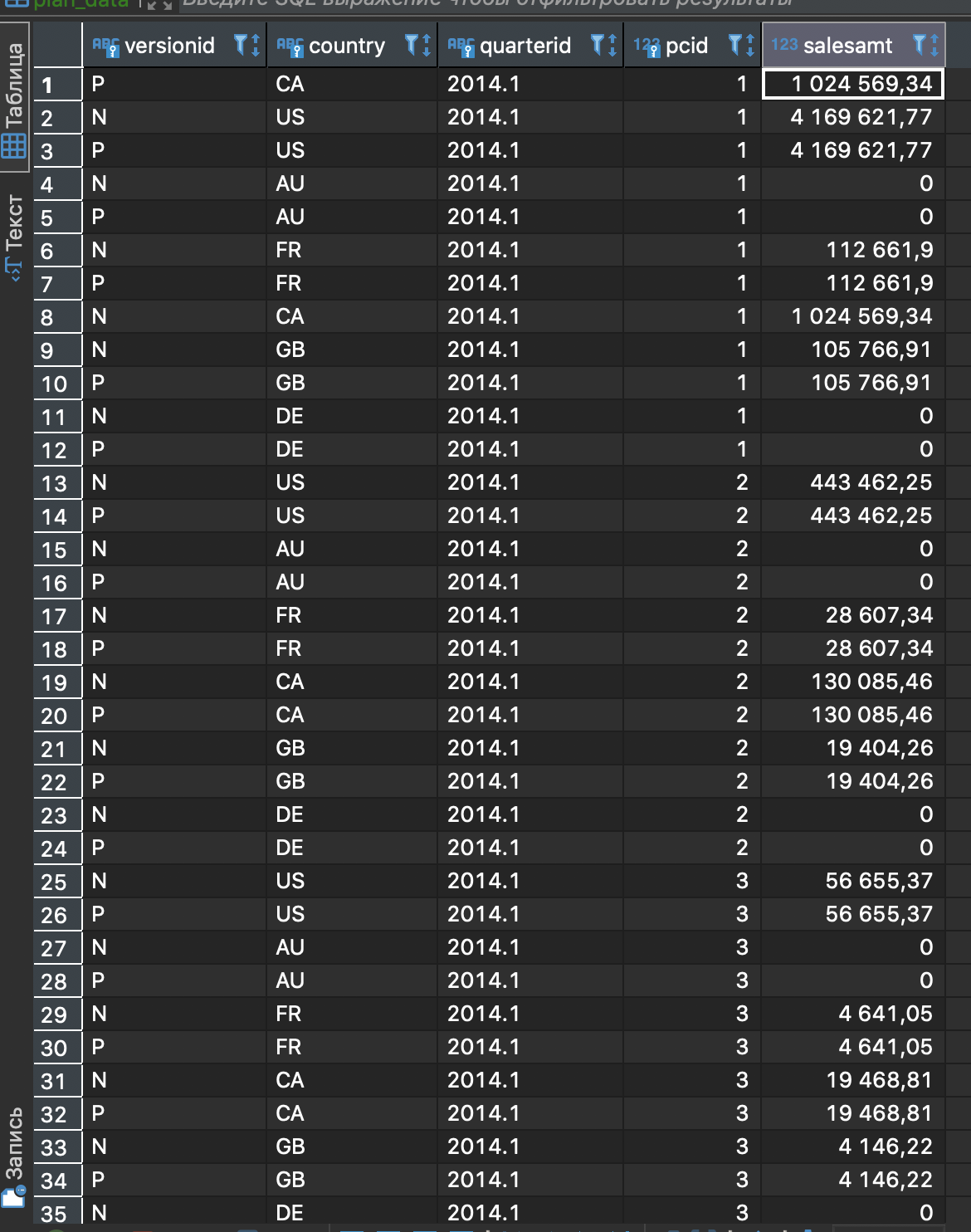
start\_planning(2014, 1, 'sophie')Result

Result

### plan\_status

### 

### plan\_data



# Task №7. Changing plan data

## Listing

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

conn = psycopg2.connect(database='planning', user=user, password=pwd, host='localhost')

cur = conn.cursor()

timestamp = datetime.datetime.now().timestamp()

current\_quarter = get\_quarter\_id(quarter, year)

cur.execute(

"""UPDATE plan\_status

SET status = 'L', modifieddatetime = to\_timestamp(%s), author = %s

FROM country\_managers

WHERE country\_managers.country = plan\_status.country and

country\_managers.username = current\_user and

quarterid = %s""",

[timestamp, user, current\_quarter]

)

conn.commit()

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

conn = psycopg2.connect(database='planning', user=user, password=pwd, host='localhost')

cur = conn.cursor()

timestamp = datetime.datetime.now().timestamp()

current\_quarter = get\_quarter\_id(quarter, year)

cur.execute(

"""UPDATE plan\_status

SET status = 'R', modifieddatetime = to\_timestamp(%s), author = %s

FROM country\_managers

WHERE country\_managers.country = plan\_status.country and

country\_managers.username = current\_user and

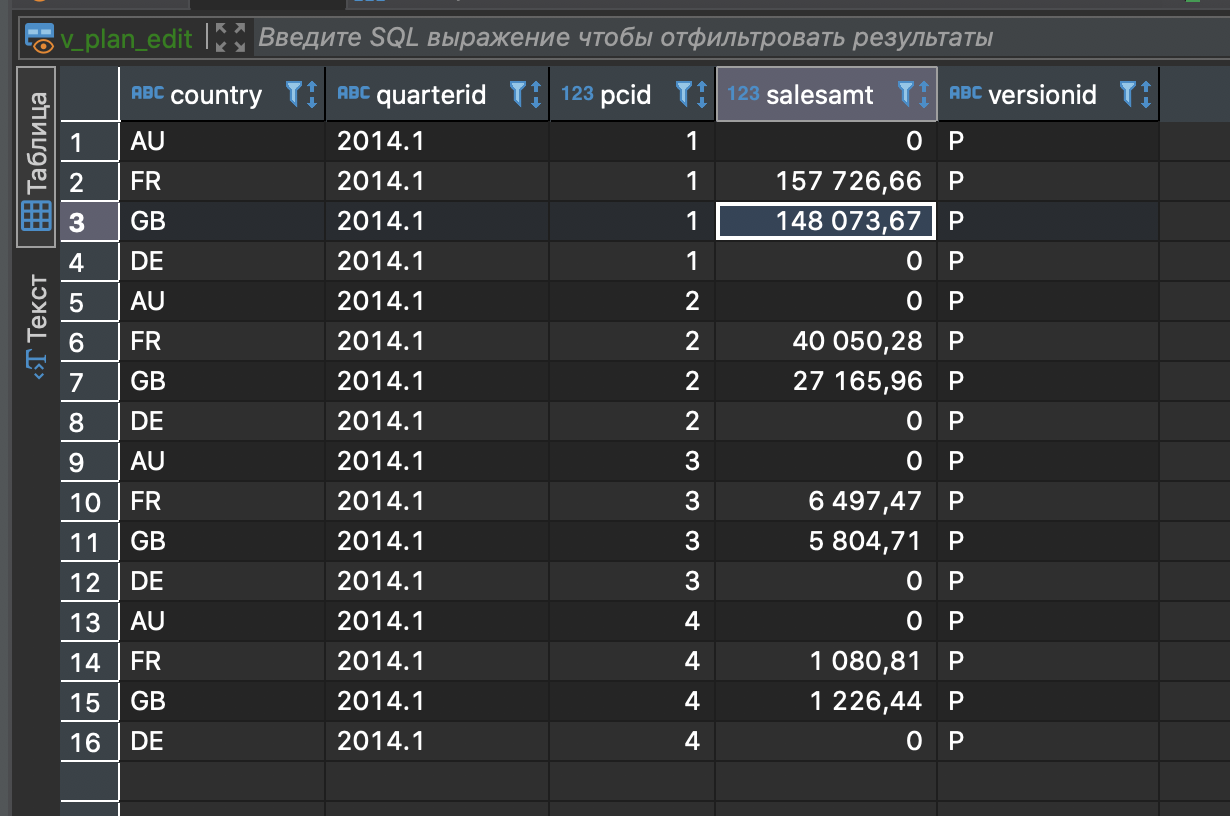
quarterid = %s""",

[timestamp, user, current\_quarter]

)

conn.commit()

## Screenshot



# Task №8. Plan data approval

## Listing

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

conn = psycopg2.connect(database='planning', user=user, password=pwd, host='localhost')

cur = conn.cursor()

timestamp = datetime.datetime.now().timestamp()

current\_quarter = get\_quarter\_id(quarter, year)

new\_plan\_qr = """select pd.versionid, pd.country, pd.quarterid, pd.pcid, salesamt from plan\_data pd

join plan\_status ps on pd.country = ps.country and pd.quarterid = ps.quarterid

join country\_managers cm on pd.country = cm.country

where pd.quarterid = %s and versionid = 'P' and ps.status = 'R' and cm.username = current\_user"""

cur.execute(new\_plan\_qr, [current\_quarter])

new\_plan\_records = cur.fetchall()

cur.execute('select country from country\_managers where username = current\_user')

current\_user\_countries = tuple([item[0] for item in cur.fetchall()])

delete\_old\_plan\_qr = """delete from plan\_data

where versionid = 'A' and quarterid = %s and country in %s"""

cur.execute(delete\_old\_plan\_qr, [current\_quarter, (current\_user\_countries)])

conn.commit()

for record in new\_plan\_records:

cur.execute("""INSERT INTO plan\_data VALUES ('A', %s, %s, %s, %s)""", [\*record[1:]])

conn.commit()

update\_plan\_status\_qr = """UPDATE plan\_status

SET status = 'A', modifieddatetime = to\_timestamp(%s)

WHERE quarterid = %s and author = current\_user and country in %s"""

timestamp = datetime.datetime.now().timestamp()

cur.execute(update\_plan\_status\_qr, [timestamp, current\_quarter, (current\_user\_countries)])

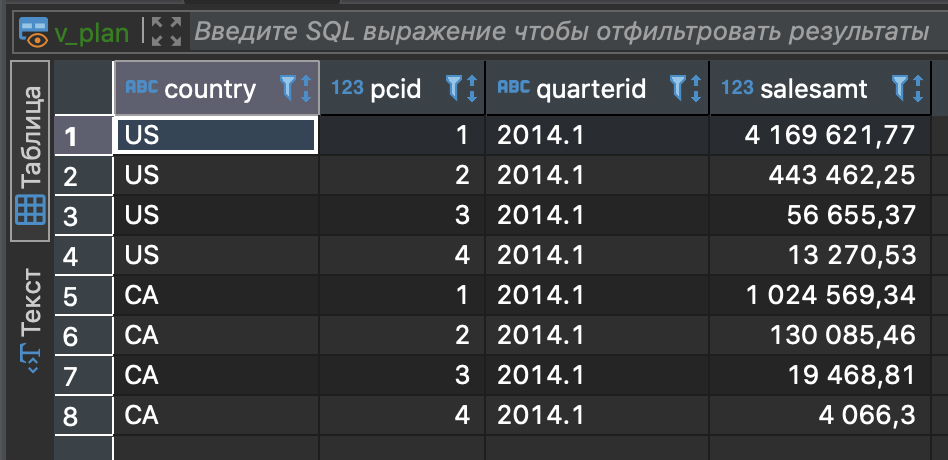
conn.commit()

## Function calls

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

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

## Screenshot



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

## Approach

Calculate actual data using salesorderheader and ordersalesdetail tables without using company\_sales.

## SQL query

**drop** **materialized** view **if** **exists** mv\_plan\_fact\_2014\_q1;

**drop** **table** **if** **exists** temp\_6;

**drop** **table** **if** **exists** temp\_5;

**drop** **table** **if** **exists** temp\_4;

**drop** **table** **if** **exists** temp\_3;

**drop** **table** **if** **exists** temp\_2;

**drop** **table** **if** **exists** temp\_1;

**select** customerid,

sod.linetotal,

sod.productid,

**date\_part**('year', soh.orderdate) **as** "year",

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

**date\_part**('year', soh.orderdate) || '.' || **date\_part**('quarter', soh.orderdate) **as** qr

**into** temp\_1

**from** salesorderdetail sod

**inner** **join** salesorderheader soh

**on** sod.salesorderid = soh.salesorderid;

**select** t1.\*, cc.cid

**into** temp\_2

**from** temp\_1 t1

**inner** **join** (

**select** c2.id **as** cid,

c.customerid

**from** customer c

**inner** **join** company c2

**on** c.companyname = c2.cname

) **as** cc

**on** t1.customerid = cc.customerid

**where** "year" = 2014;

**select** t2.\*,

p2.pcid,

p2.pcname **as** category\_name

**into** temp\_3

**from** temp\_2 t2

**inner** **join** product2 p2

**on** t2.productid = p2.productid;

**select** t3.\*,

ccls.cls

**into** temp\_4

**from** temp\_3 t3

**inner** **join** (

**select** \*

**from** company\_abc ccls

**where** ccls."year" = 2013 **and** ccls.cls **in** ('A','B')

) **as** ccls

**on** t3.cid = ccls.cid ;

**select** t4.\*,

countrycode

**into** temp\_5

**from** temp\_4 t4

**inner** **join** company c

**on** t4.cid = c.id;

**select** countrycode,

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

qr,

pcid **as** categoryid,

category\_name

**into** temp\_6

**from** temp\_5

**group** **by** qr, countrycode , pcid, category\_name, "year", quarter\_yr;

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

**select** **distinct** quarterid,

country,

category\_name,

**NULLIF**(pd.salesamt, 0) - t6.salesamt **as** dev,

(pd.salesamt - t6.salesamt)/**NULLIF**(pd.salesamt, 0) **as** devpercent

**from** temp\_6 t6 **join** plan\_data pd

**on** t6.countrycode = pd.country **and** t6.qr = pd.quarterid **and** t6.categoryid = pd.pcid;

## Screenshot

