НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ І УПРАВЛІННЯ

Комп’ютерний практикум №2

з дисципліни

“ OLAP та сховища даних”

Виконала:

студентка групи ІС-71

Дмитрук О.Ю.

Перевірив:

старший викладач

Олійник Ю.О.

Київ-2019

1. Показати ранг кожного товару у групі/категорії (відповідно до зменшення ціни в групі). Запит реалізувати через функції rank() та dense\_rank(). Порівняти результати виконання. **Крім того сформуйте запит без використання аналітичних функцій.**

Rank()

select

product\_name,

price,

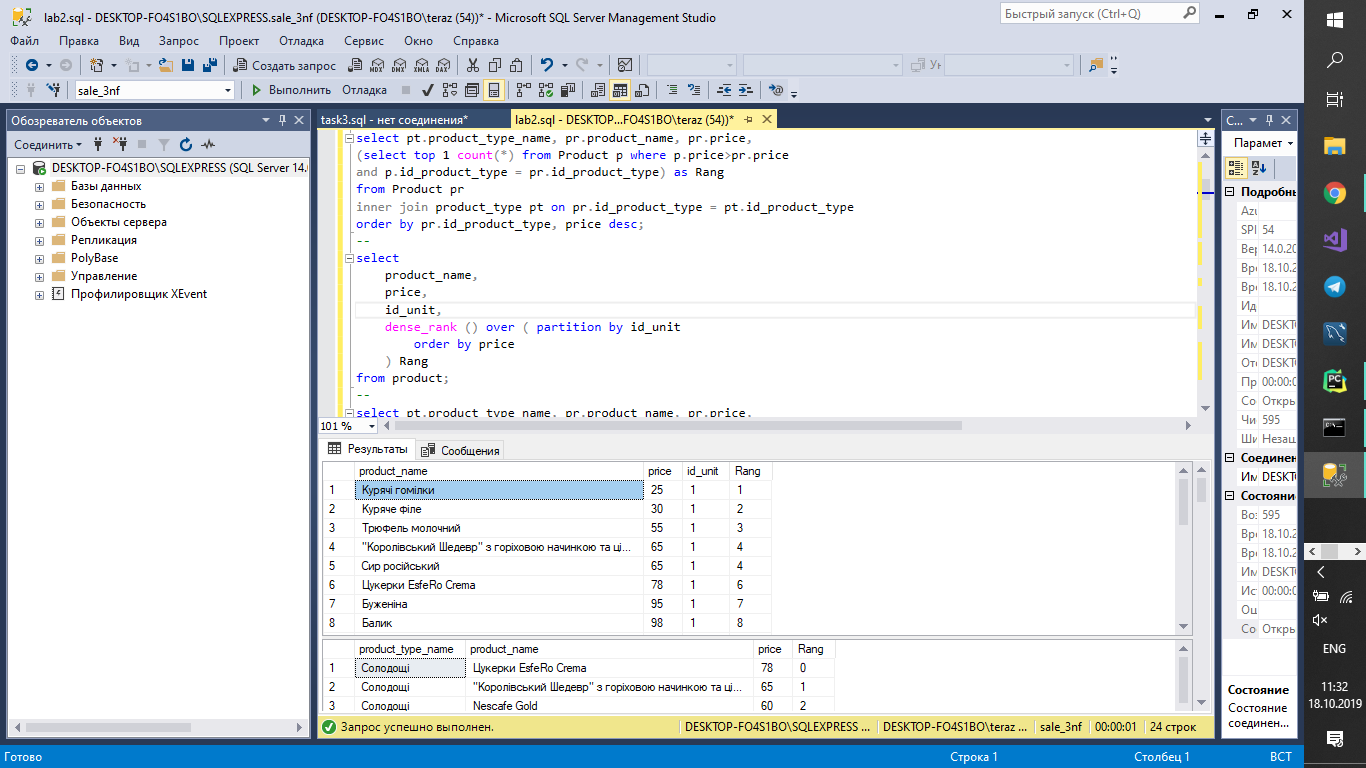
id\_unit,

rank () over ( partition by id\_unit

order by price

) Rang

from product;



*Рисунок 1*

Без аналітичної функції Rank() (але з аналітичною функцією Count())

select pt.product\_type\_name, pr.product\_name, pr.price,

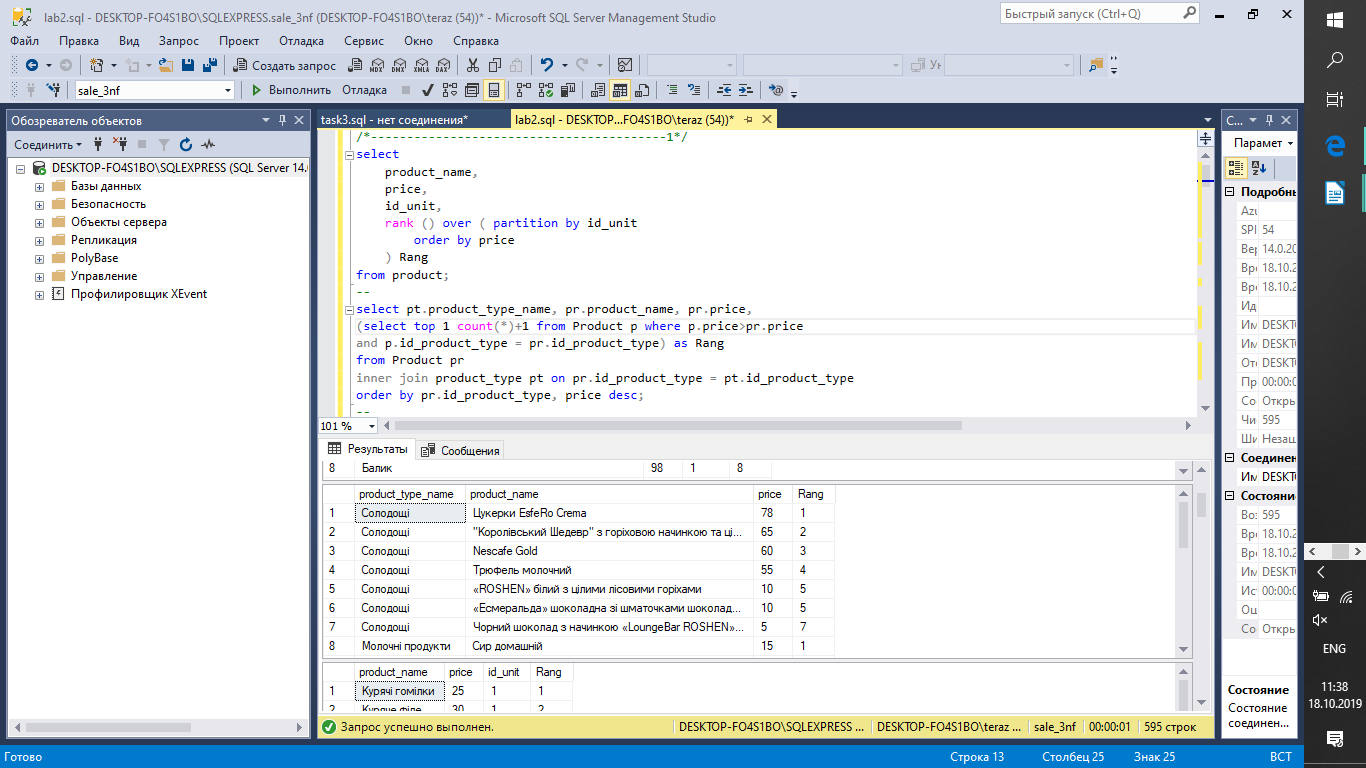
(select top 1 count(\*)+1 from Product p where p.price>pr.price

and p.id\_product\_type = pr.id\_product\_type) as Rang

from Product pr

inner join product\_type pt on pr.id\_product\_type = pt.id\_product\_type

order by pr.id\_product\_type, price desc;



*Рисунок 2*

Dense\_rank()

select

product\_name,

price,

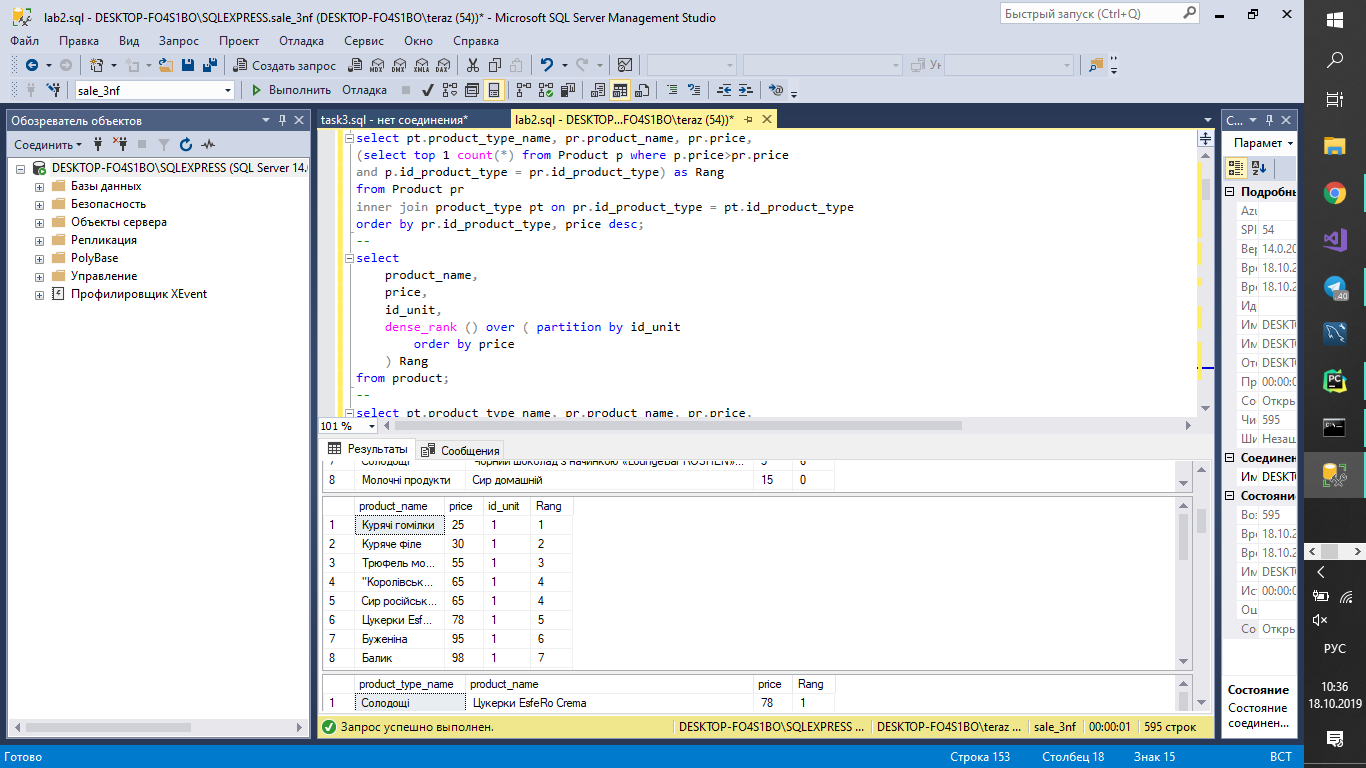
id\_unit,

dense\_rank () over ( partition by id\_unit

order by price

) Rang

from product;



*Рисунок 3*

Без аналітичної функції Dense\_rank() (але з аналітичною функцією Count())

select *pt.product\_type\_name, pr.product\_name, pr.price,*

(select top 1 count(\*)+1 from (

select id\_product\_type, price

from product

group by id\_product\_type, price) p

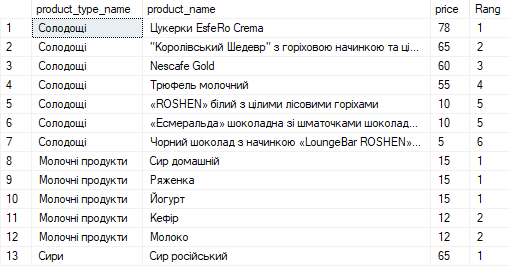
where p.price > pr.price and p.id\_product\_type = pr.id\_product\_type

) as Rang

from product pr

inner join product\_type pt on pr.id\_product\_type=pt.id\_product\_type

*order by pr.id\_product\_type, price desc;*



*Рисунок 4*

Обидва запити виводять рейтинг у впорядкованій групі. Через функцію rank() зв’язки присвоюються однаковому рангу, а наступний рейтинг пропускається. При використанні dense\_rank() , ніякі ранги не пропущені.

2. За допомогою аналітичного SQL сформуйте запит для виведення списку із трьох найдешевших товарів у кожній групі. Крім того сформуйте запит без використання аналітичних функцій.

Row\_number()

select *\* from (*

select pt.product\_type\_name, p.product\_name, p.price, p.id\_unit,

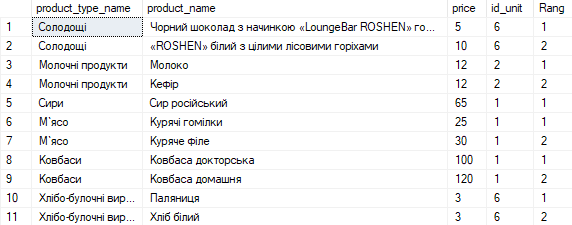
row\_number() over (partition by p.id\_product\_type order by price) as Rang

from product p

inner join product\_type pt on p.id\_product\_type=pt.id\_product\_type

) as min3

*where min3.Rang<3;*



*Рисунок 5*

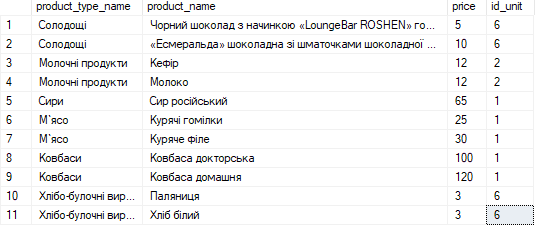
Без аналітичної функції Row\_number()

select pt.product\_type\_name, p.product\_name, p.price, p.id\_unit

from product\_type pt

cross apply (select top 2 \* from product

*where pt.id\_product\_type = product.id\_product\_type order by price) p;*



*Рисунок 6*

4. Показати, які товари по кожній групі мають найбільші та найменші продажі. Крім того сформуйте запит без використання аналітичних функцій.

З використанням Rank()

with task\_result as(

select sum(s.quantity) as sales\_queantity,p.id\_product, p.id\_product\_type

from store s

inner join product p on s.id\_product=p.id\_product group by p.id\_product, p.id\_product\_type,s.id\_product),

task\_4 as (

select r.id\_product,

r.id\_product\_type,

sales\_queantity, rank() over (partition by r.id\_product\_type order by sales\_queantity desc) as rank\_

from task\_result r group by r.id\_product, r.id\_product\_type, r.sales\_queantity)

select p.id\_product,p.product\_name, pt.product\_type\_name, t.sales\_queantity, t.rank\_

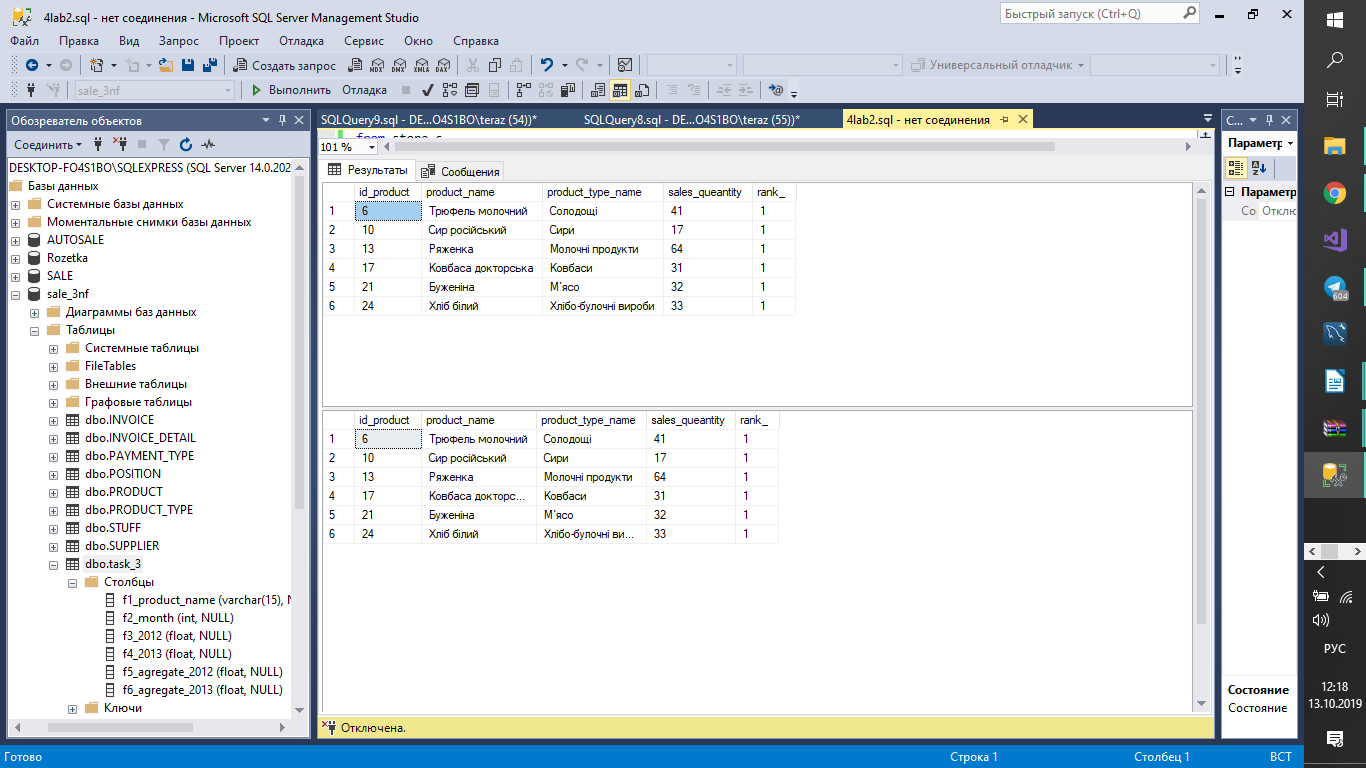
from task\_4 t

inner join product p on p.id\_product=t.id\_product

inner join product\_type pt on pt.id\_product\_type=t.id\_product\_type

where rank\_=1

order by t.id\_product ;



*Рисунок 7*

Без аналітичної функції Rank() (але з аналітичною функцією Count())

with *task\_result as(select sum(s.quantity) as sales\_queantity,p.id\_product, p.id\_product\_type*

from store s

inner join product p on s.id\_product=p.id\_product group by p.id\_product, p.id\_product\_type,s.id\_product

),

task\_4 as (select r.id\_product,

r.id\_product\_type,

sales\_queantity,(

select 1 + count(\*)

from task\_result task\_result\_

where task\_result\_.id\_product\_type = r.id\_product\_type and

task\_result\_.sales\_queantity > r.sales\_queantity)

as rank\_

from task\_result r group by r.id\_product, r.id\_product\_type, r.sales\_queantity)

select p.id\_product,p.product\_name, pt.product\_type\_name, t.sales\_queantity, t.rank\_

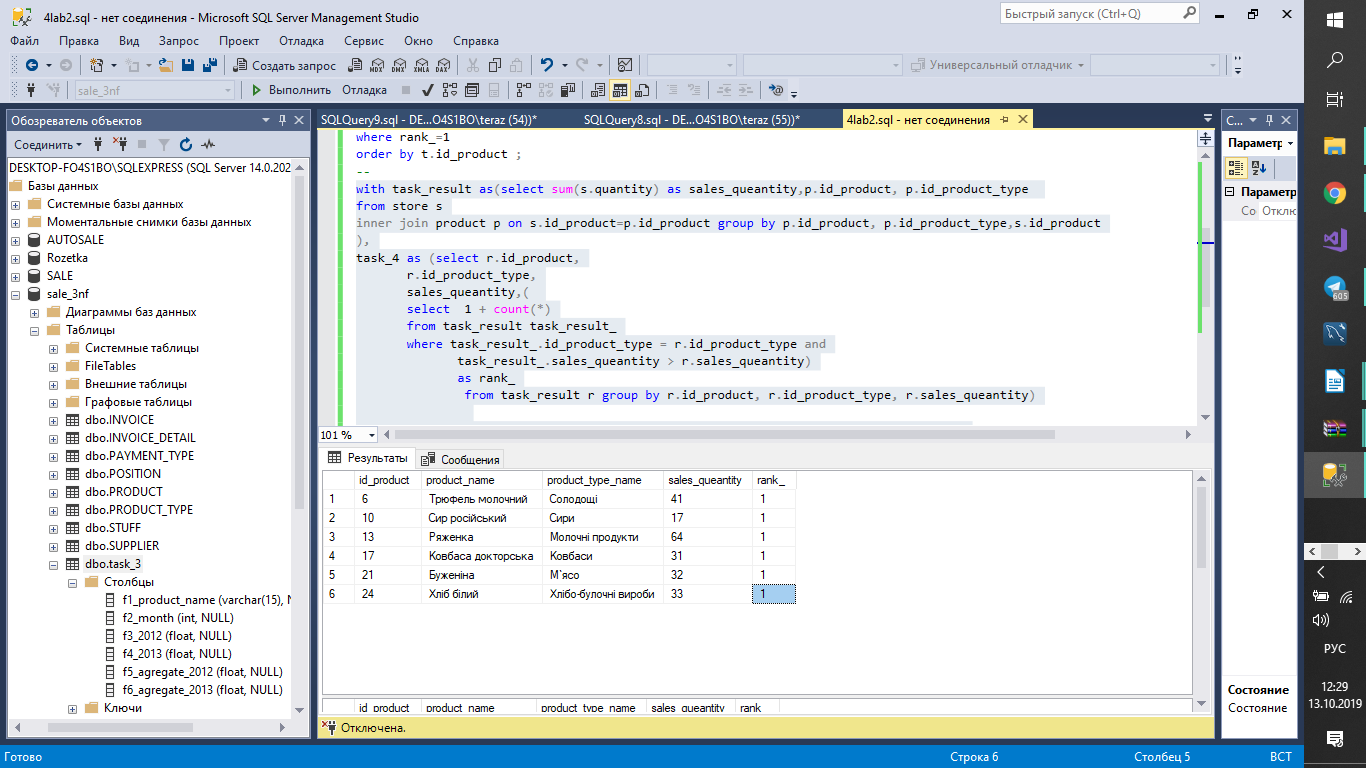
from task\_4 t

inner join product p on p.id\_product=t.id\_product

inner join product\_type pt on pt.id\_product\_type=t.id\_product\_type

where rank\_=1

*order by t.id\_product;*



*Рисунок 8*

5. По кожному товару підрахувати кількість товарів, у яких вартість вища від даного товару у діапазоні від 5 до 10 включно. Наприклад якщо прайс товару 6, то знайти кількість товарів, у яких діапазон прайсів від 11 до 16.

select *pr.product\_name, t.count*

from product pr

inner join (

select pf.id\_product, count (\*) as [count]

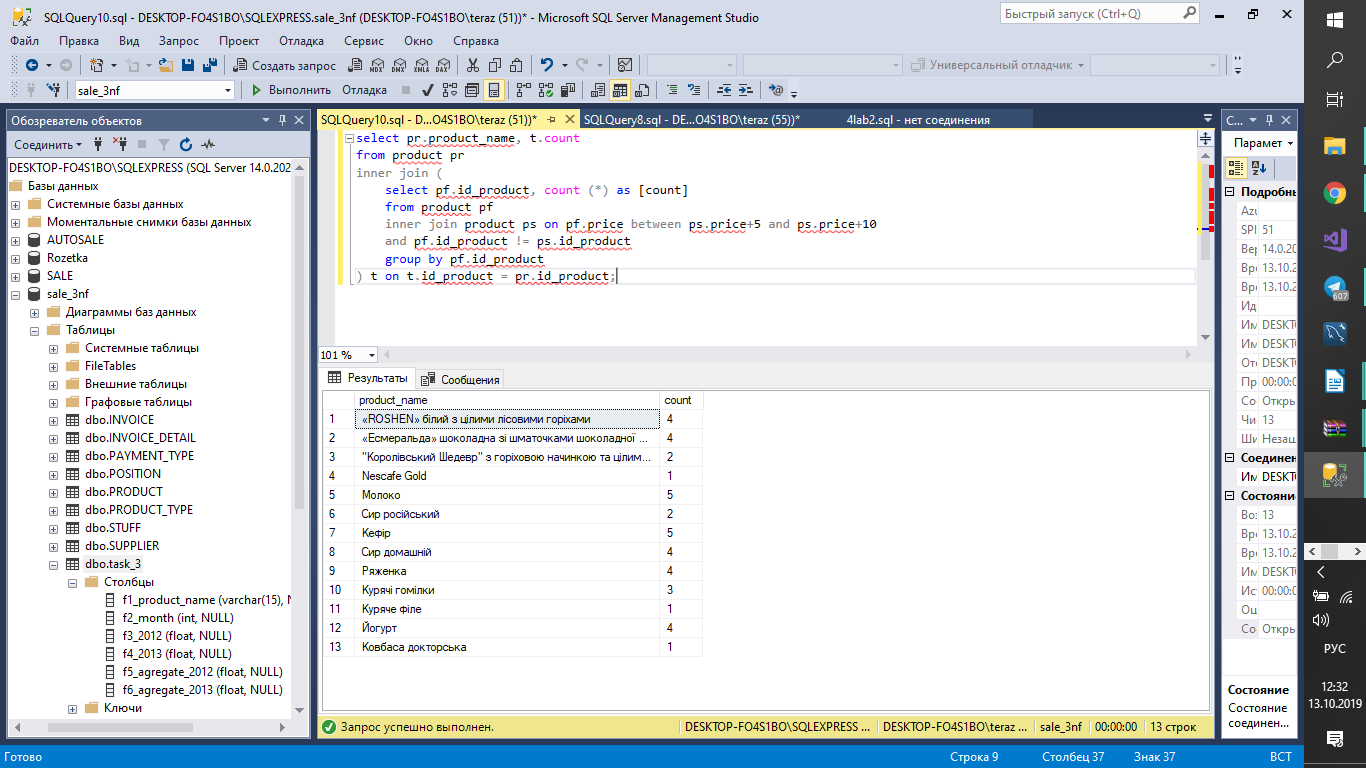
from product pf

inner join product ps on pf.price between ps.price+5 and ps.price+10

and pf.id\_product != ps.id\_product

group by pf.id\_product

*) t on t.id\_product = pr.id\_product;*



*Рисунок 9*

SELECT id\_product, product\_name,

count(\*) over (order by price rows between 5 following and 10 following) count

from product p

order by id\_product;

6. По кожному товару вивести першу(f3) та останню(f4) дату продажів по кожному місяцю, cуму продажів (f5), відсоток від річної суми(f6).

use sale\_3nf;

with first\_last as(

select distinct p.id\_product, p.product\_name,month(s.date\_oper) as mounth,

min(s.date\_oper) over (partition by month(s.date\_oper),year(s.date\_oper), p.id\_product) AS first\_date,

max(s.date\_oper) over (partition by month(s.date\_oper),year(s.date\_oper), p.id\_product) AS last\_date,

sum(s.quantity\*p.price) over( partition by p.id\_product, month(s.date\_oper),year(s.date\_oper)

) as summa,

sum(s.quantity\*p.price) over( partition by p.id\_product,year(s.date\_oper)

) as year\_price

from product p

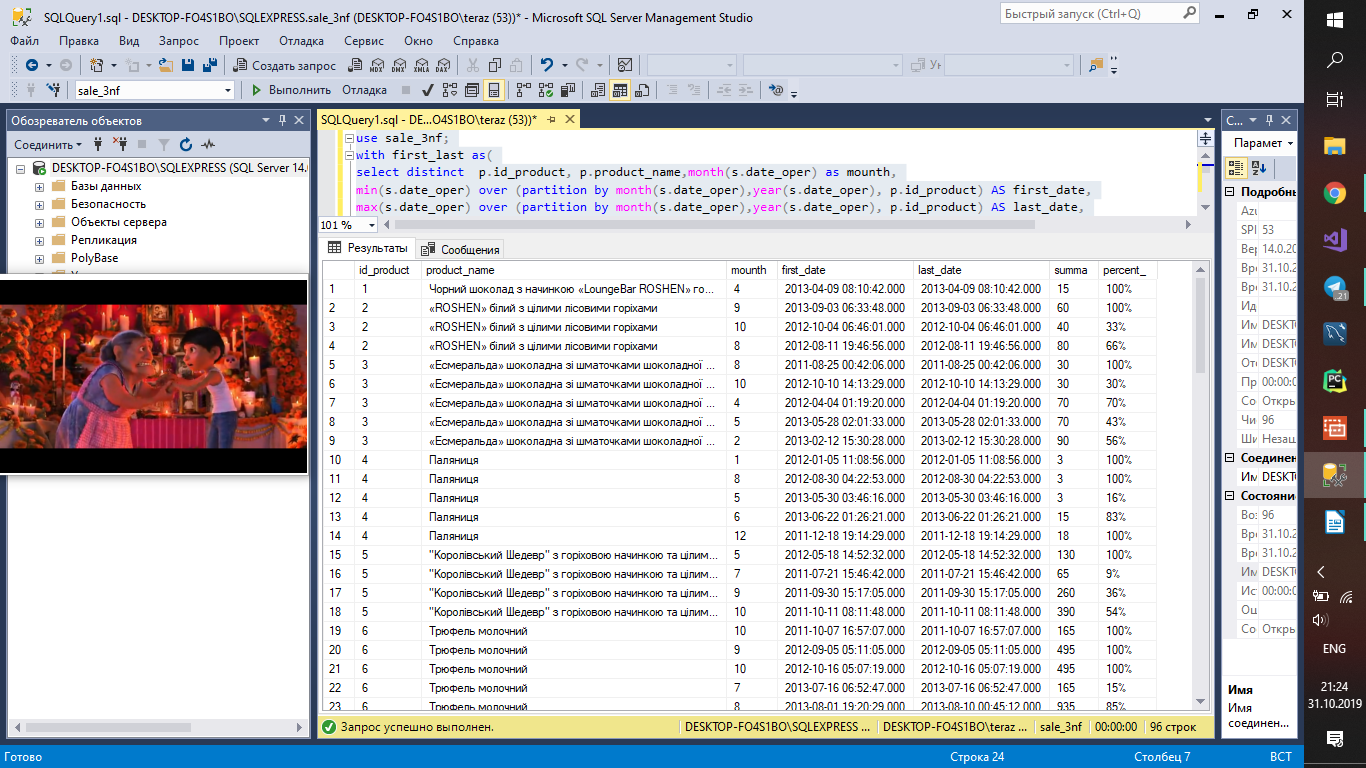
inner join store s on p.id\_product=s.id\_product

group by p.id\_product,p.product\_name,s.date\_oper,s.quantity\*p.price)

select

id\_product, product\_name, mounth, first\_date, last\_date,summa,

concat((sum(summa\*100/year\_price) over(partition by id\_product,year\_price,summa)),'%') as percent\_ from first\_last;



*Рисунок 10*

7. За допомогою ROLLUP та CUBE виведіть проміжні суми продажу по групам товарів.

ROLLUP

select isnull (pr.product\_name, 'ALL') as 'Prod\_name', p.date, p.amount

from (

select sto.id\_product, isnull(convert(nvarchar, date\_oper), 'ALL') as [date], sum(quantity\*price) as [amount]

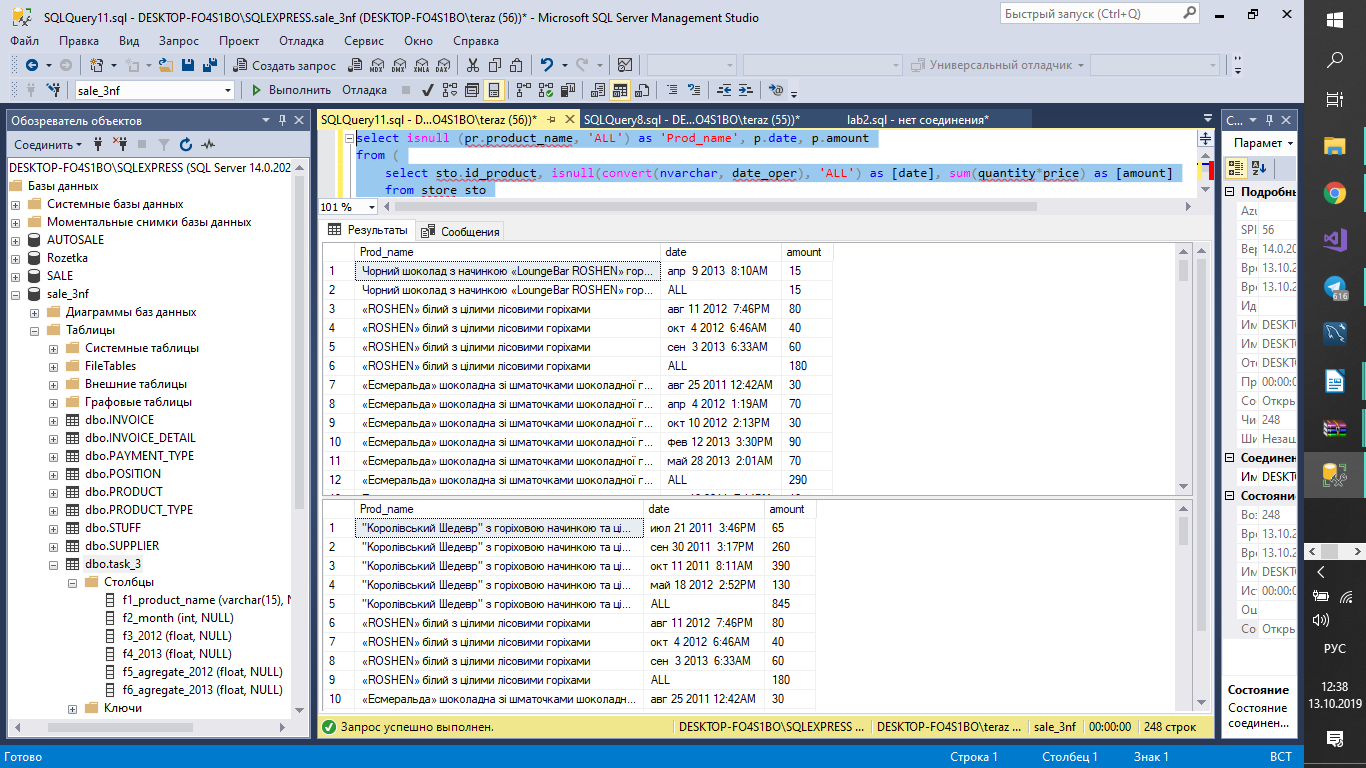
from store sto

inner join product pr on pr.id\_product=sto.id\_product

group by rollup (sto.id\_product, date\_oper)

)as p

left join product pr on pr.id\_product=p.id\_product;



*Рисунок 11*

CUBE

select isnull (pr.product\_name, 'ALL') as 'Prod\_name', p.date, p.amount

from (

select sto.id\_product, isnull(convert(nvarchar, date\_oper), 'ALL') as [date], sum(quantity\*price) as [amount]

from store sto

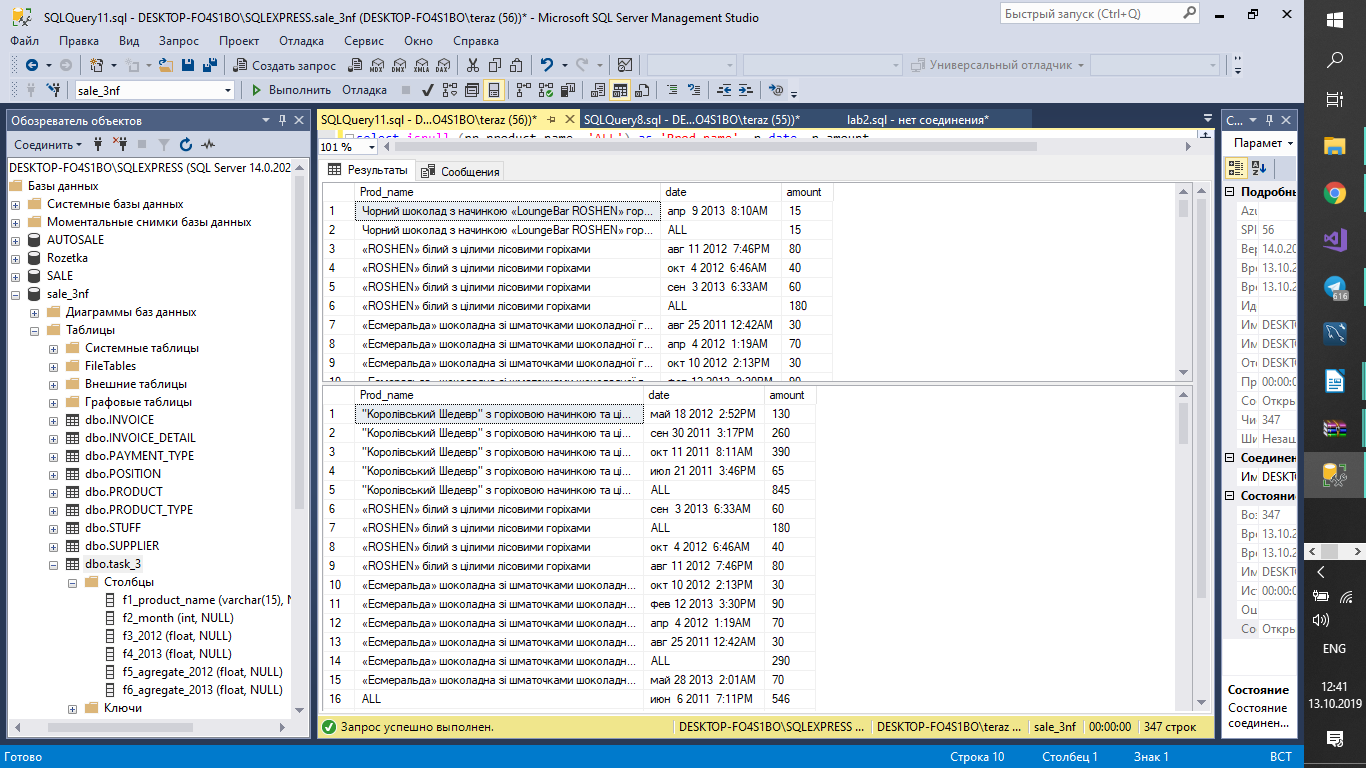
inner join product pr on pr.id\_product=sto.id\_product

group by cube (sto.id\_product, date\_oper)

)as p

left join product pr on pr.id\_product=p.id\_product

order by prod\_name;



*Рисунок 12*

3. F1 – продукти по яким здійснюються продажі, f2 – номер місяця, f3 – продажі по товару за певний місяць 2012 року, f4 – продажі по товару за певний місяць 2013 року, f5 – наростаючий підсумок продажів по товару за певний місяць 2012 року, f5 – наростаючий підсумок продажів по товару за певний місяць 2013 року.

Крім того сформуйте запит без використання аналітичних функцій.

select f1\_product\_name, f2\_month, f3\_2012,f4\_2013,

sum(f3\_2012) over(order by f2\_month

rows between unbounded preceding

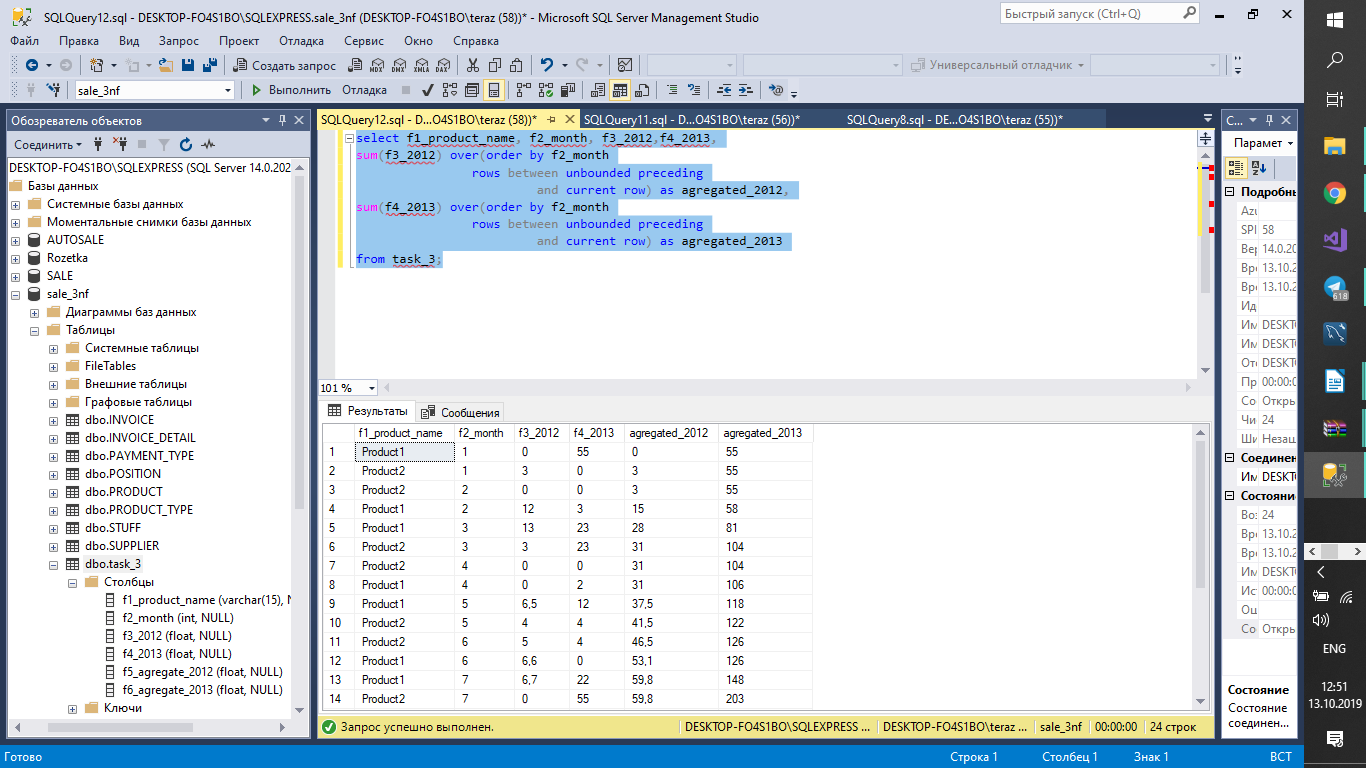
and current row) as agregated\_2012,

sum(f4\_2013) over(order by f2\_month

rows between unbounded preceding

and current row) as agregated\_2013

from task\_3;



*Рисунок 13*

with result as

(select id\_product, cast(year(purchase\_time) as nvarchar) year,

cast(month(purchase\_time) as nvarchar) month, quantity from invoice i inner join invoice\_detail id ON i.id\_invoice = id.id\_invoice)

select distinct f1.id\_product, f1.month,

sum(f1.quantity) over (partition by f1.id\_product, f1.month) "2012",

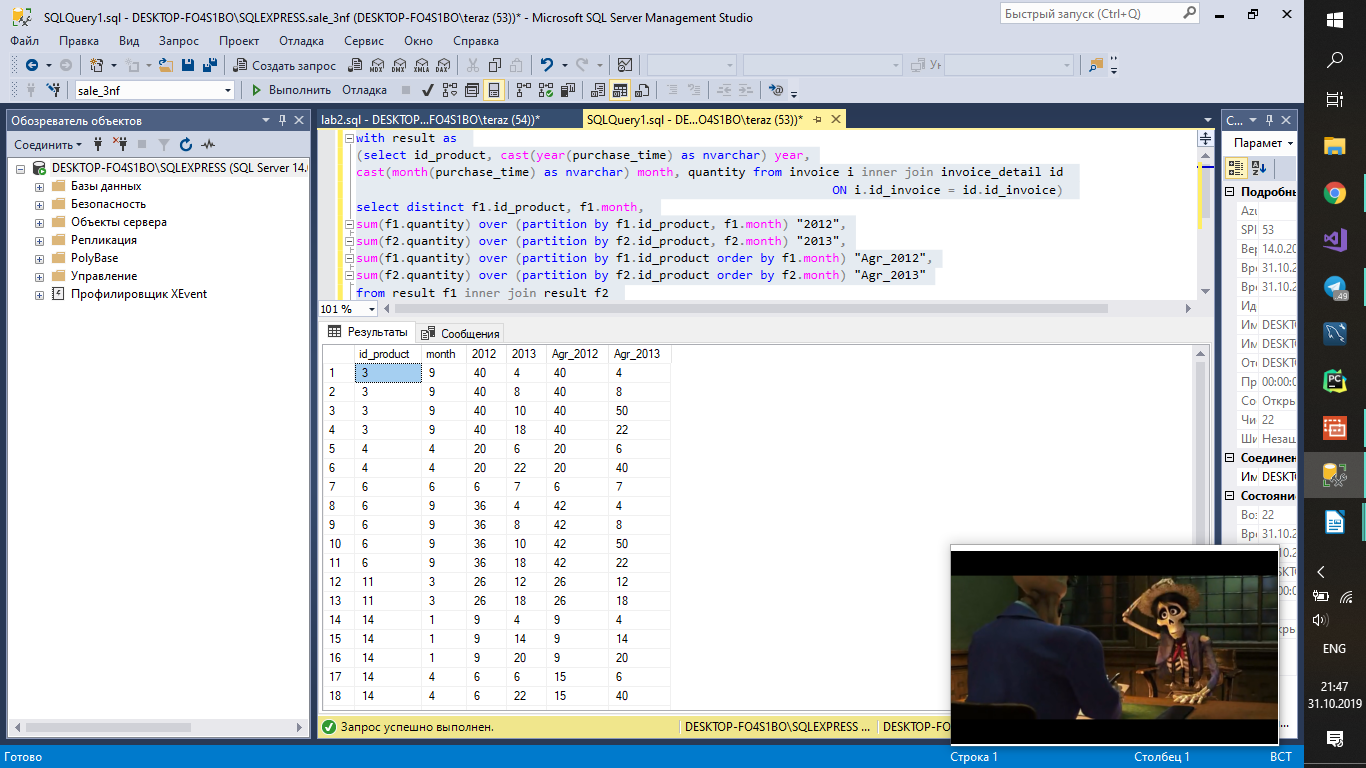
sum(f2.quantity) over (partition by f2.id\_product, f2.month) "2013",

sum(f1.quantity) over (partition by f1.id\_product order by f1.month) "Agr\_2012",

sum(f2.quantity) over (partition by f2.id\_product order by f2.month) "Agr\_2013"

from result f1 inner join result f2

on f1.month = f2.month and f1.year = 2012 and f2.year = 2013 order by f1.id\_product;



*Рисунок 14*