Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**О Т Ч Ё Т**

по лабораторной работе №2

по дисциплине:

«Информационное обеспечение систем управления»

на тему:

«Группировка и обобщение данных»

Выполнил: ст. гр. ВУЦ-421

Полунин С.К.

Вариант 6

Проверил: доц., к.т.н.

Васильева М. А.

Москва ­2024

**Цель работы**

Изучить операторы SQL диалекта Postgres Pro, необходимые для группировки и обобщения данных. Научиться создавать запросы с агрегатными функциями.

**Результаты выполненной работы**

**Задание** **1**

Формулировка задания:

Найти минимальную дату рождения из таблицы HumanResources.Employee.

Текст скрипта:

*SELEC*T **DATE\_PART**('YEAR', AGE(CURRENT\_DATE,

**max** (**"BirthDate"**))) **AS** **"MinData"**

**FROM** **"HumanResources"**.**"Employee"**

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

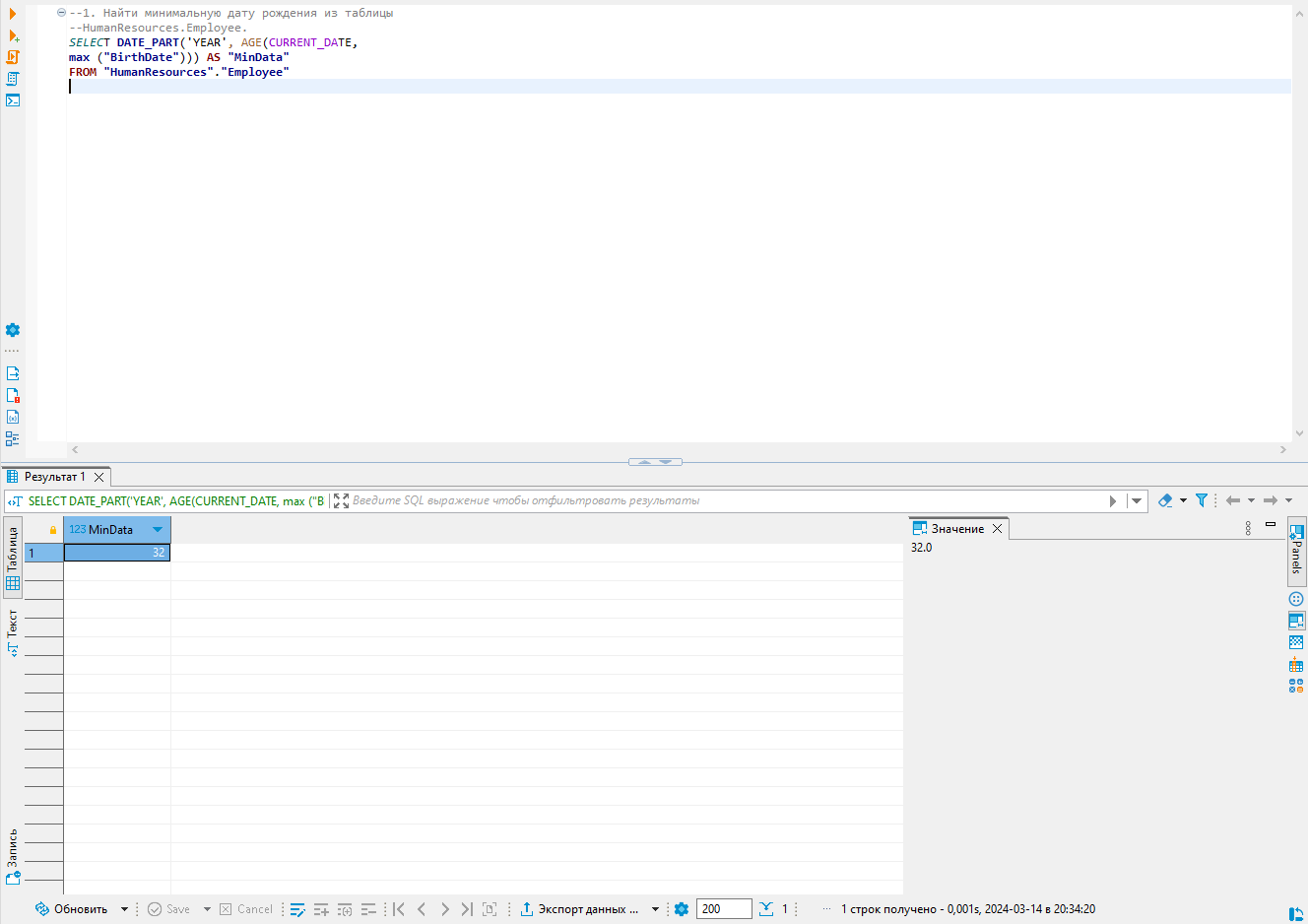


Рисунок 1 – Результат выполнения первого задания

**Задание** **2**

Формулировка задания:

Найти суммарную цену (ListPrice) товара сорокового размера и черного цвета из таблицы Production.Product.

Текст скрипта:

**select** **SUM**("ListPrice") **AS** *"SumPrice"*

**FROM** "Production"."Product"

**WHERE** "Size" = **'40'** **AND** "Color" **LIKE** **'Black%'**

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

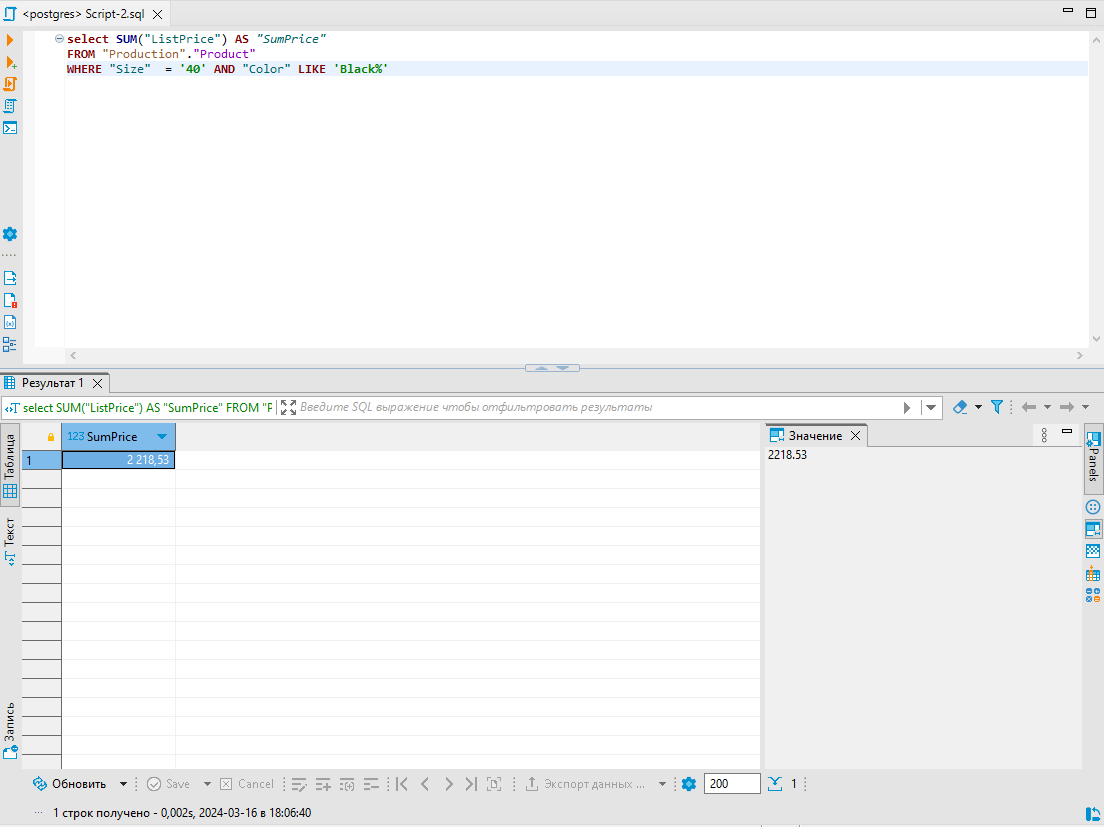


Рисунок 2 – Результат выполнения второго задания

**Задание** **3**

Формулировка задания:

Найти количество человек с именем David из таблицы HumanResources.Employee.

Текст скрипта:

**SELECT** **COUNT**(**'David'**) **AS** *"Product Count"*

**FROM** "HumanResources"."Employee"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

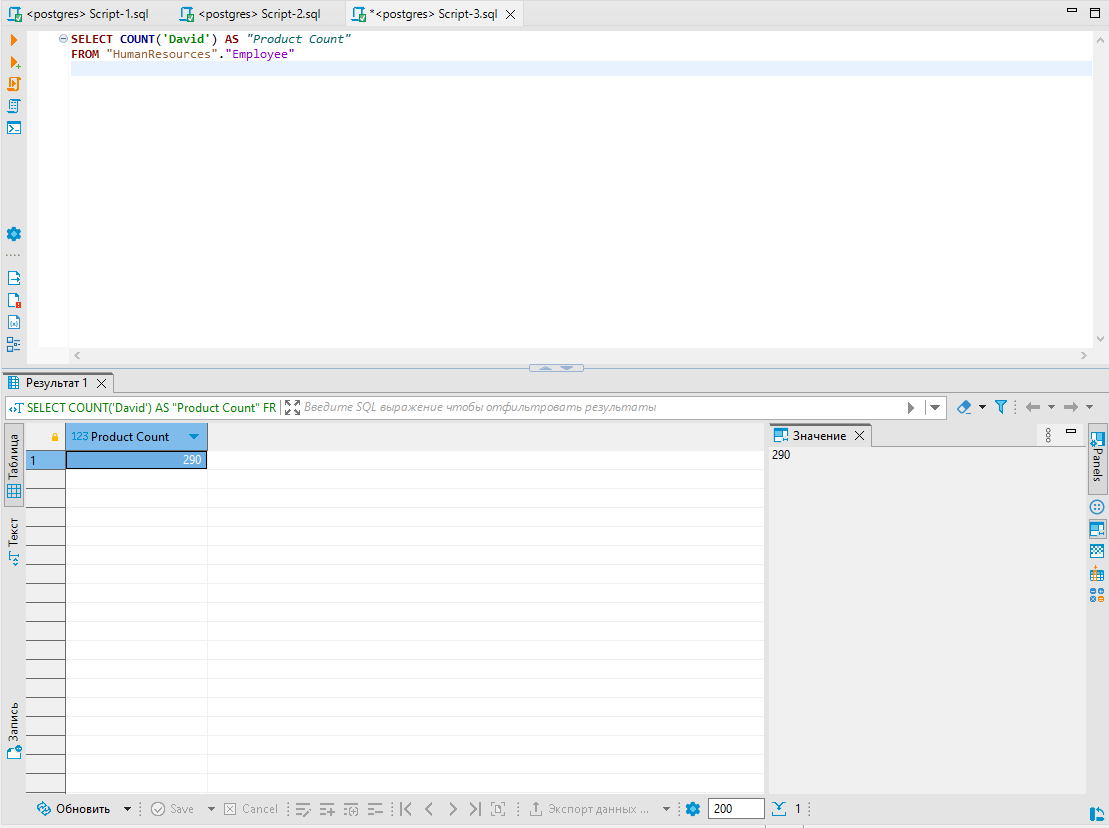


Рисунок 3 – Результат выполнения третьего задания

**Задание** **4**

Формулировка задания:

Выбрать те линейки товаров (ProductLine) из таблицы Production.Product, где максимальная цена больше 3000 (из выборки исключить пустые поля).

Текст скрипта:

**SELECT** **DISTINCT** "ProductLine"

**FROM** "Production"."Product"

**WHERE** "ListPrice" **IS** **NOT** **NULL**

**GROUP** **BY** "ProductLine"

**HAVING** **MAX**("ListPrice") > 3000

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

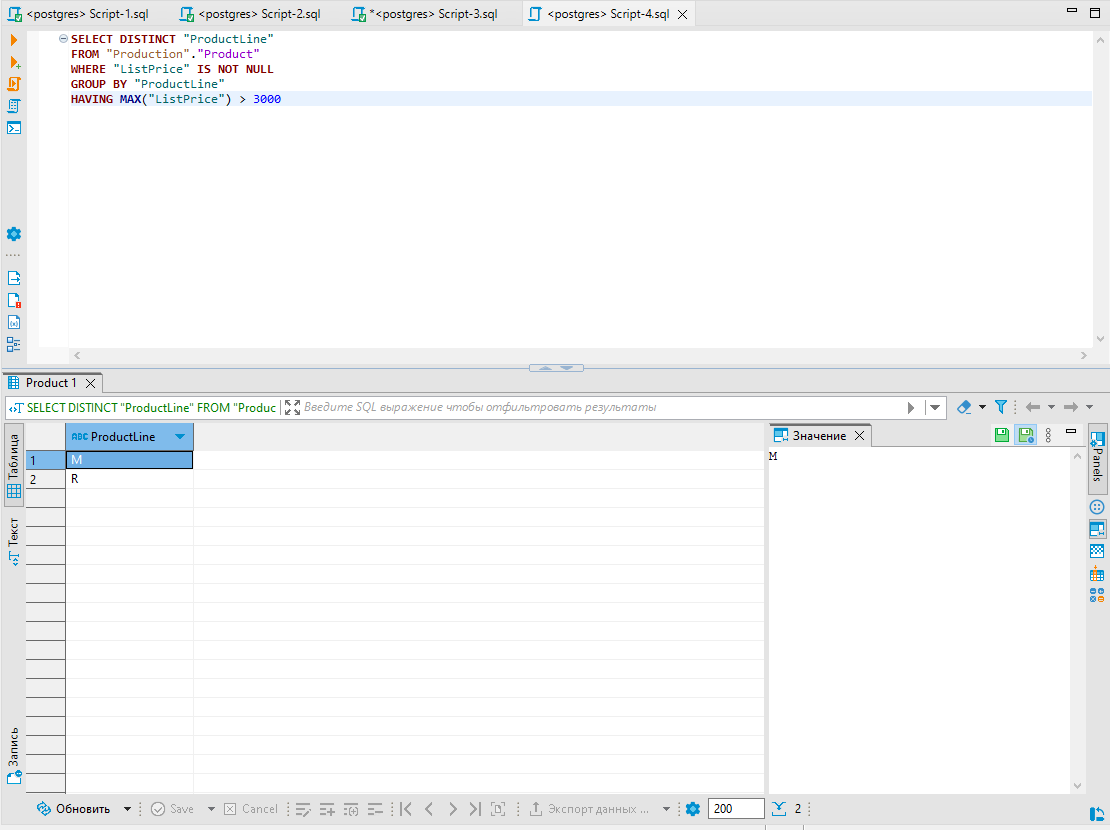


Рисунок 4 − Результат выполнения четвертого задания

**Задание** **5**

Формулировка задания:

Вывести EmployeeID, ShipDate и сумму по SubTotal для всевозможных EmployeeID с одинаковой ShipDate, предусмотреть вывод общей суммы StandardPrice за ShipDate из таблицы Purchasing.PurchaseOrderHeader. (Использовать CUBE).

Текст скрипта:

**SELECT**

"EmployeeID"

,"ShipDate"

,**SUM**("SubTotal") **AS** *"TotalSubTotal"*

**FROM** "Purchasing"."PurchaseOrderHeader"

**GROUP** **BY** **CUBE** ("EmployeeID", "ShipDate")

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

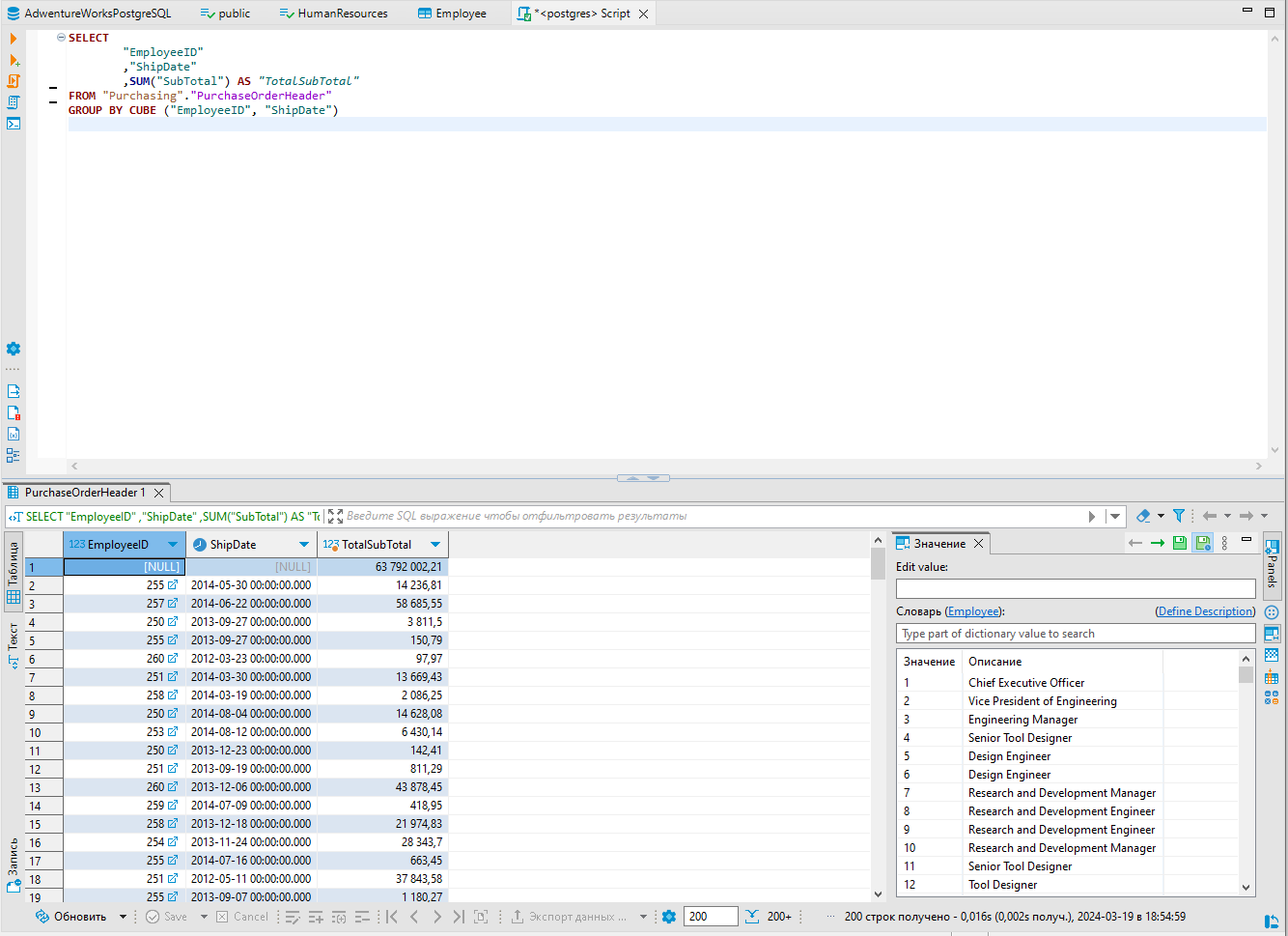


Рисунок 5 − Результат выполнения пятого задания

**Задание** **6**

Формулировка задания:

Изменить запрос п.5 использовать CUBE. Отделить строки, созданные с помощью агрегатных функций от строк из фактической таблицы.

Текст скрипта:

**SELECT**

"EmployeeID"

,"ShipDate"

,**SUM**("SubTotal") **AS** *"TotalSubTotal"*

,**GROUPING**("EmployeeID") **AS** *"Grouping"*

**FROM** "Purchasing"."PurchaseOrderHeader"

**GROUP** **BY** **CUBE** ("EmployeeID", "ShipDate")

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

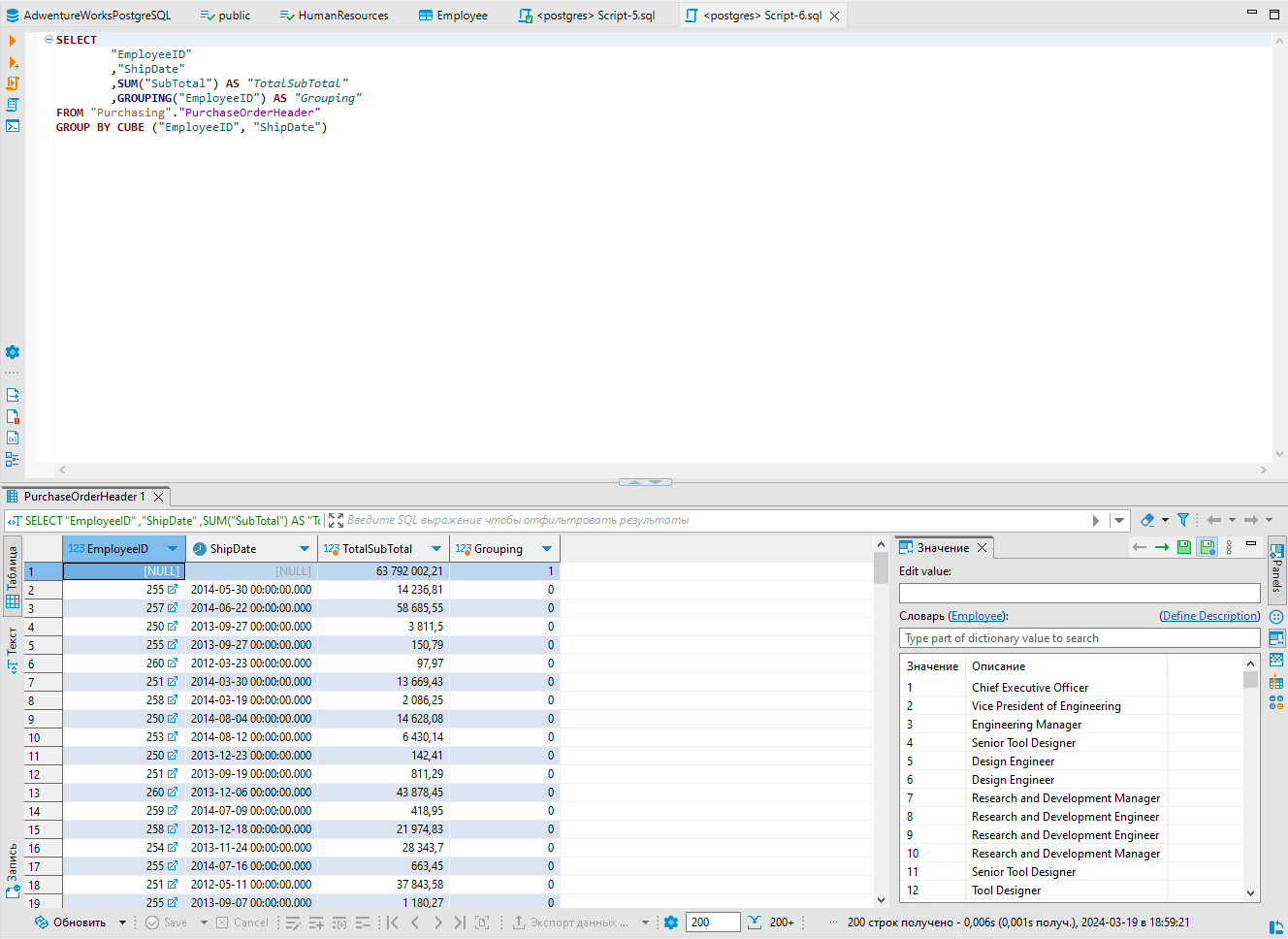


Рисунок 6 − Результат выполнения шестого задания

**Задание** **7**

Формулировка задания:

Изменить запрос п.5 использовать GROUPING SETS. Отделить строки, созданные с помощью агрегатных функций от строк из фактической таблицы

Текст скрипта:

**SELECT**

"EmployeeID"

,"ShipDate"

,**SUM**("SubTotal") **AS** *"TotalSubTotal"*

,**GROUPING**("EmployeeID") **AS** *"Grouping"*

**FROM** "Purchasing"."PurchaseOrderHeader"

**GROUP** **BY** **GROUPING** **SETS**("EmployeeID", "ShipDate")

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

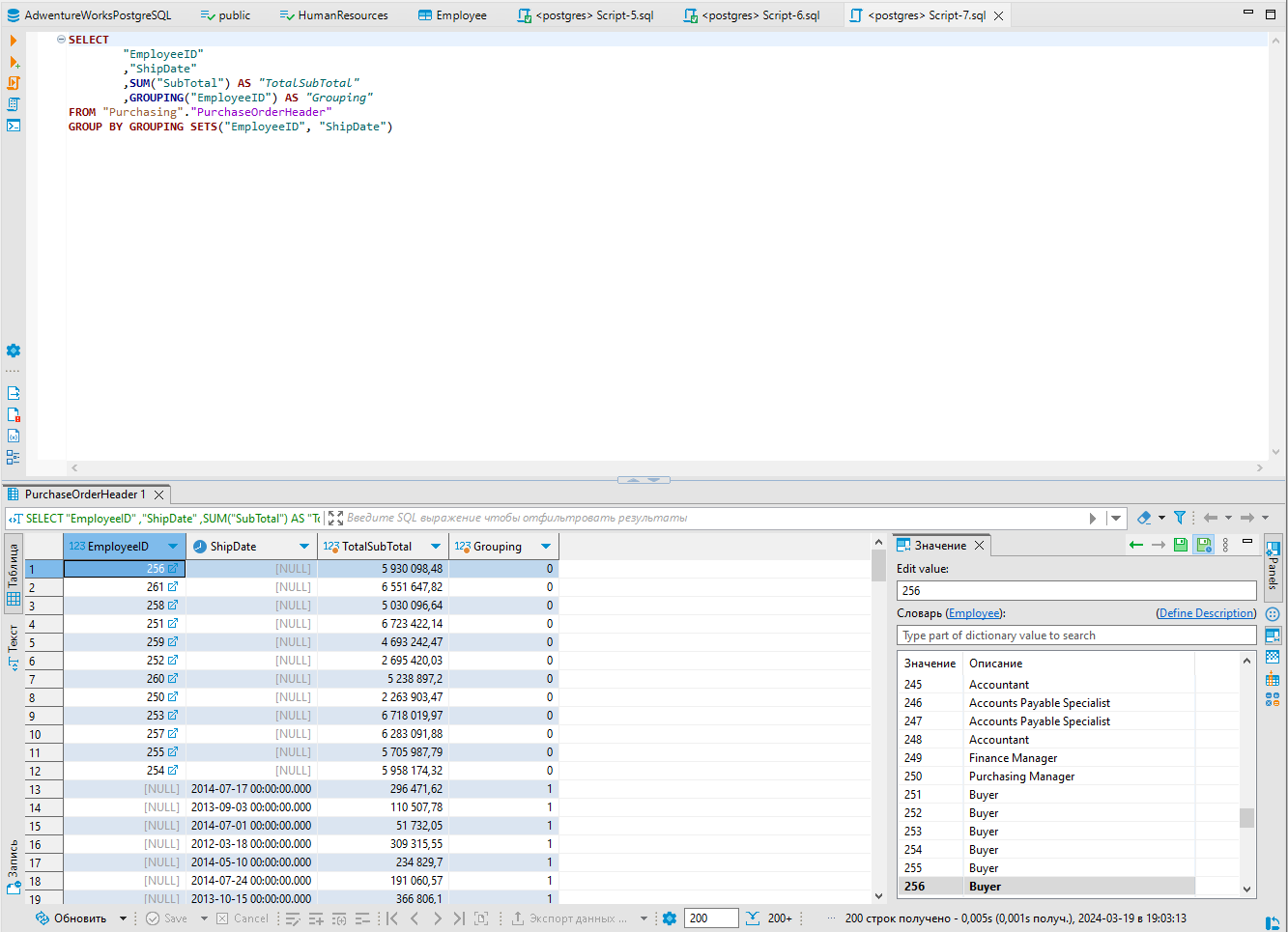


Рисунок 7 − Результат выполнения седьмого задания

Вывод: изучены операторы SQL диалекта Postgres Pro, необходимые для группировки и обобщения данных, а также научились создавать запросы с агрегатными функциями.