ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ЛАБОРАТОРНОЙ РАБОТЕ №2

По дисциплине «Информационное обеспечение систем управления»

ВАРИАНТ 5

Выполнил: ст. гр. ВУЦ - 451

Лихачев Сергей Святославович

Проверил: к.т.н., доц. Васильева М. А.

Москва 2024

## Цель работы

Изучить операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для фильтрации набора данных. Научиться создавать простые запросы на фильтрацию данных. Подготовить отчет, выполненный согласно требованиям ГОСТ 7.32−2017.

## РЕЗУЛЬТАТЫ ВЫПОЛНЕННОЙ РАБОТЫ

Задание

Формулировка задания:

Показать количество разнообразных цветов товара (Color) из

таблицы Production.Product.

Текст скрипта:

**SELECT**

"Color",

**count**("Color")

**FROM**

"Production"."Product" p

**GROUP** **BY**

p."Color"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

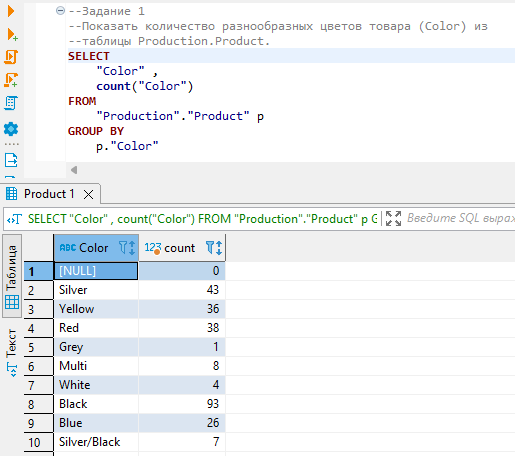


Рисунок 1 – Результат выполнения первого задания

Задание

Формулировка задания:

Найти суммарную цену (ListPrice) товара размера M из таблицы Production.Product, не учитывать пустые и нулевые значения

Текст скрипта:

**SELECT**

p."ProductID",

p."Name",

**sum**(p."ListPrice") **AS** "Суммарная стоимость"

**FROM**

"Production"."Product" p

**WHERE**

p."Size" = 'M'

**AND** p."ListPrice" **IS** **NOT** **NULL**

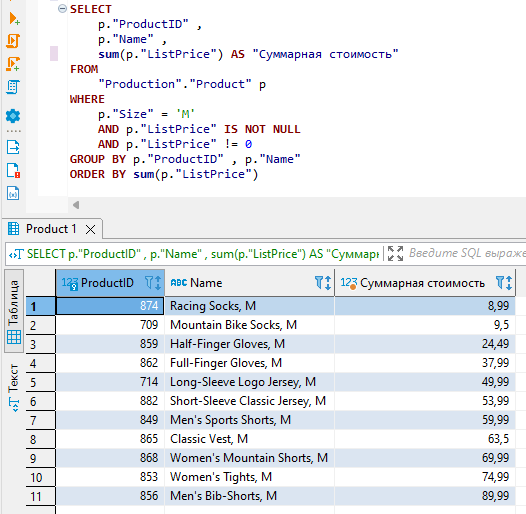
**AND** p."ListPrice" != 0

**GROUP** **BY** p."ProductID" , p."Name"

**ORDER** **BY** **sum**(p."ListPrice")

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).



**Рисунок 2 – Результат выполнения второго задания**

Задание 3

Формулировка задания:

Найти количество номеров телефона разного типа (PhoneNumberTypeID) из таблицы Person.PersonPhone, где номер начинается с 819. Сортировать по возрастанию.

Текст скрипта:

**SELECT**

pnt."Name" **AS** "Тип" ,

**count**(pp."PhoneNumber") **AS** "Количество"

**FROM**

"Person"."PersonPhone" pp

**JOIN** "Person"."PhoneNumberType" pnt

**ON** pnt."PhoneNumberTypeID" = pp."PhoneNumberTypeID"

**WHERE**

pp."PhoneNumber" **LIKE** '819%'

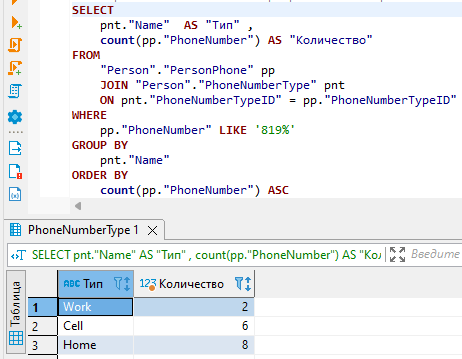
**GROUP** **BY**

pnt."Name"

**ORDER** **BY**

**count**(pp."PhoneNumber") **ASC**

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3). 

**Рисунок 3 – Результат выполнения третьего задания**

Задание 4

Формулировка задания:

Вывести пол, где средний возраст больше 42 лет из таблицы HumanResources.Employee (не учитывать пустые значения).

Текст скрипта:

**SELECT**

e."Gender" **AS** "Пол",

**EXTRACT** (years **FROM** **avg**(**age**(**current\_date** , e."BirthDate"))) **AS** "Средний возраст"

**FROM**

"HumanResources"."Employee" e

**WHERE**

e."BirthDate" **IS** **NOT** **NULL**

**GROUP** **BY**

e."Gender"

**HAVING**

**avg**(**age**(**current\_date** , e."BirthDate")) >= **INTERVAL** '42 years'

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

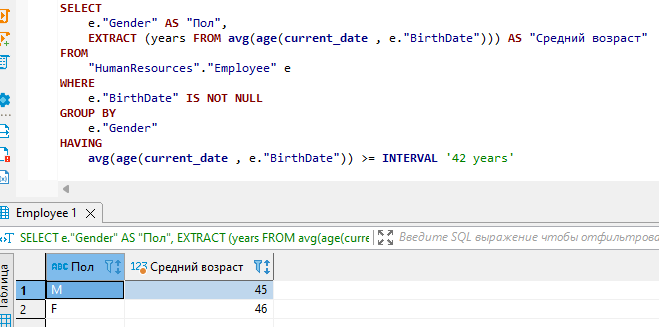


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

Формулировка задания:

Вывести EmployeeID, ShipDate и среднее по SubTotal для всевозможных EmployeeID с одинаковой ShipDate из таблицы Purchasing.PurchaseOrderHeader. (Использовать ROLLUP)

Текст скрипта:

**SELECT**

poh."EmployeeID" ,

poh."ShipDate" ,

**avg**(poh."SubTotal")

**FROM**

"Purchasing"."PurchaseOrderHeader" poh

**WHERE**

poh."EmployeeID" **IS** **NOT** **NULL**

**AND** poh."ShipDate" **IS** **NOT** **NULL**

**GROUP** **BY**

**ROLLUP** (poh."EmployeeID" , poh."ShipDate")

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

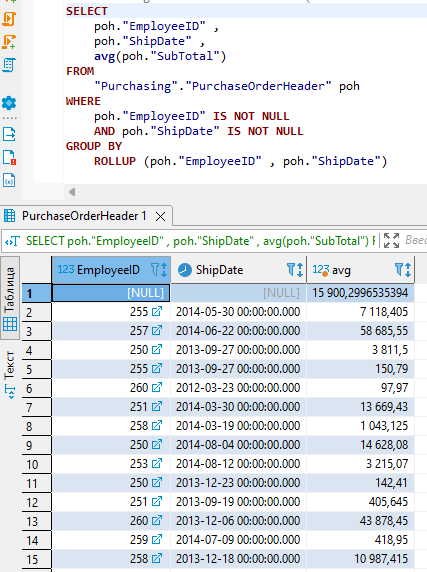


Рисунок 5 − Результат выполнения пятого задания

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**

poh."EmployeeID" ,

poh."ShipDate" ,

**avg**(poh."SubTotal"),

**GROUPING**(poh."EmployeeID") **AS** "1",

**GROUPING**(poh."ShipDate") **AS** "2"

**FROM**

"Purchasing"."PurchaseOrderHeader" poh

**GROUP** **BY**

**CUBE** (poh."EmployeeID" , poh."ShipDate")

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

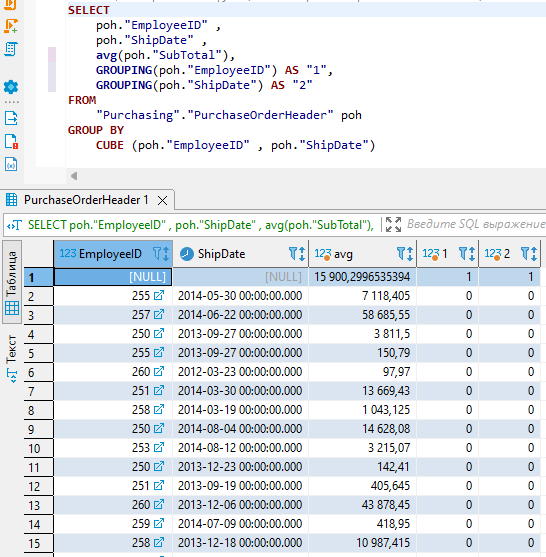


Рисунок 6 − Результат выполнения шестого задания

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**

poh."EmployeeID",

poh."ShipDate" ,

**AVG**(poh."SubTotal"),

**GROUPING**(poh."EmployeeID") **AS** "1",

**GROUPING**(poh."ShipDate") **AS** "2"

**FROM**

"Purchasing"."PurchaseOrderHeader" poh

**GROUP** **BY**

**GROUPING** **SETS**((poh."EmployeeID"), (poh."ShipDate"),())

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

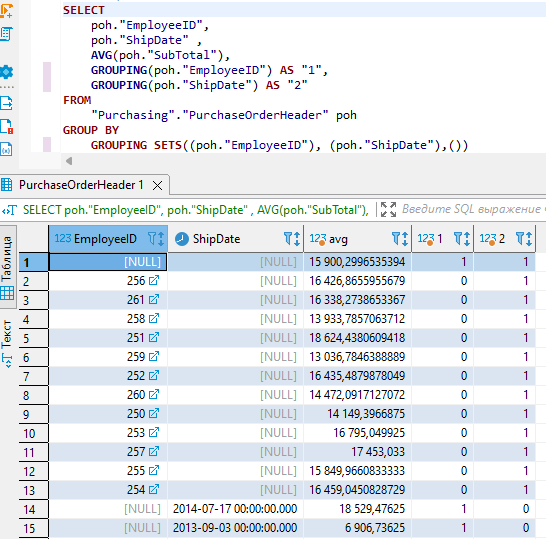


Рисунок 7 − Результат выполнения седьмого задания

## Вывод

В результате выполнения лабораторной работы изучили операторы SQL диалекта Postgres Pro, необходимые для группировки и обобщения данных. Научились создавать запросы с агрегатными функциями.