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 «Основы построения защищенных баз данных»

ВАРИАНТ 6

Выполнил: ст. гр. ТКИ-441

Комаричев Георгий Юрьевич

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

Москва 2024

## Цель работы

Изучить операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для группировки и обобщения данных. Научиться создавать запросы с агрегатными функциями. Подготовить отчет, выполненный согласно требованиям ГОСТ 7.32−2017.

## РЕЗУЛЬТАТЫ ВЫПОЛНЕННОЙ РАБОТЫ

Задание 1

1. Формулировка задания:

Найти минимальную дату рождения из таблицы HumanResources.Employee.

1. Текст скрипта:

--Найти минимальную дату рождения из таблицы HumanResources.Employee.

**SELECT** **MIN**("BirthDate") **as** *"Oldest"*

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

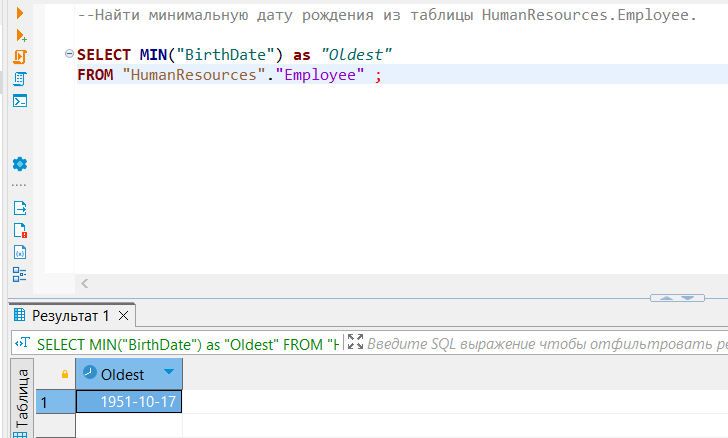


Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

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

1. Текст скрипта:

--Найти суммарную цену (ListPrice) товара сорокового размера и

--черного цвета из таблицы Production.Product

**SELECT** **SUM**("ListPrice") **as** *"SumPrice"*

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

**WHERE** "Size" **like** **'40%'** **AND** "Color" **like** **'Black%'**;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

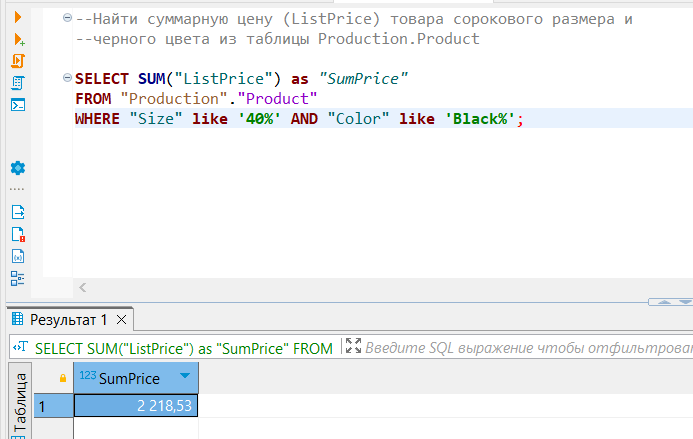


Рисунок 2 − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

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

1. Текст скрипта:

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

**SELECT** **COUNT**( \* ) **as** *"Name Count"*

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

**where** "LoginID" **like** **'%david%'**;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

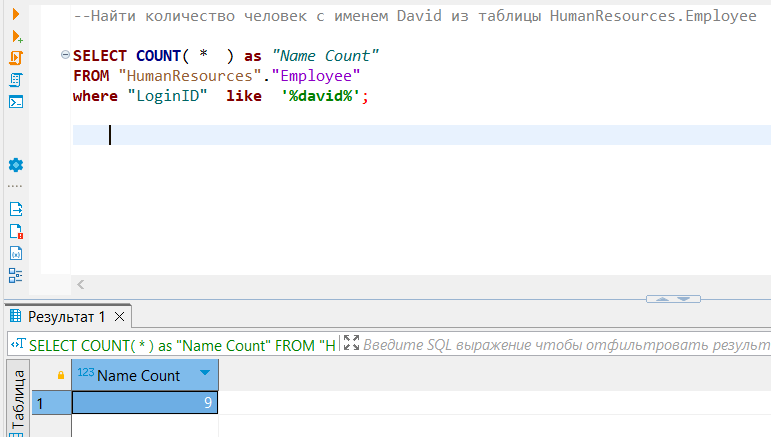


Рисунок 3 − Результат выполнения третьего задания

Задание 4

1. Формулировка задания:

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

1. Текст скрипта:

--Выбрать те линейки товаров (ProductLine) из таблицы

--Production.Product, где максимальная цена больше 3000 (из выборки исключить пустые поля).

**SELECT**

"ProductLine"

, **MAX**("ListPrice") **AS** *"MaxPrice"*

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

**WHERE** "ProductLine" **IS** **NOT** **NULL**

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

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

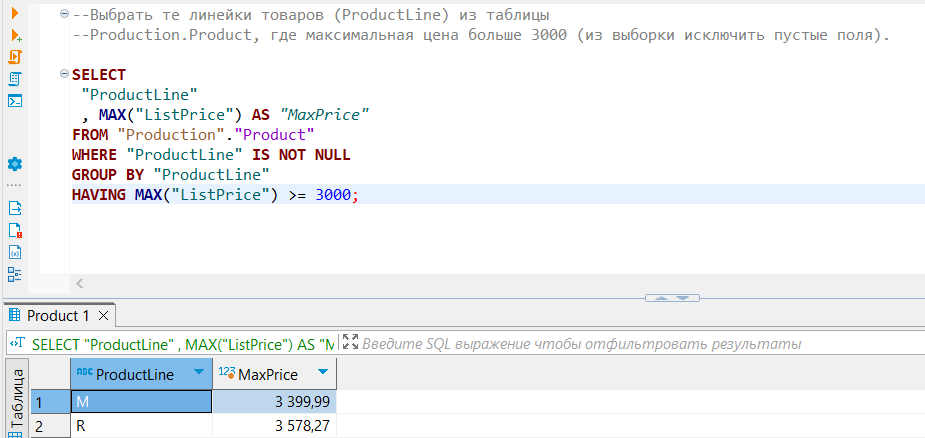


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

1. 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).

1. Текст скрипта:

--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).

**SELECT**

"EmployeeID"

, "ShipDate"

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

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

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

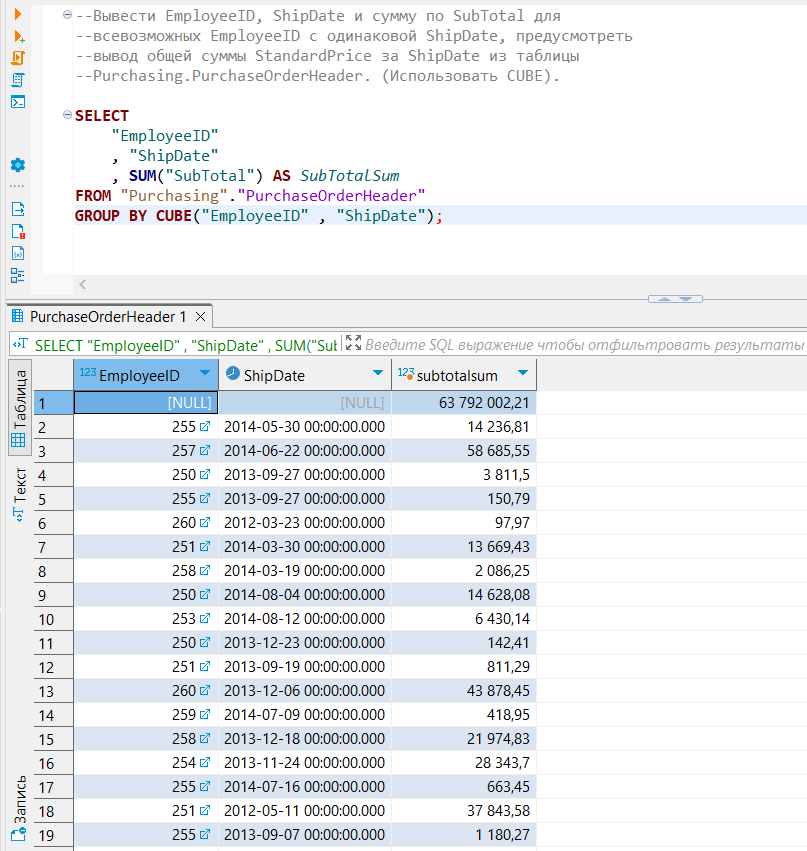


Рисунок 5 − Результат выполнения пятого задания

Задание 6

1. Формулировка задания:

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

1. Текст скрипта:

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

**SELECT**

"EmployeeID"

, "ShipDate"

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

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

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

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

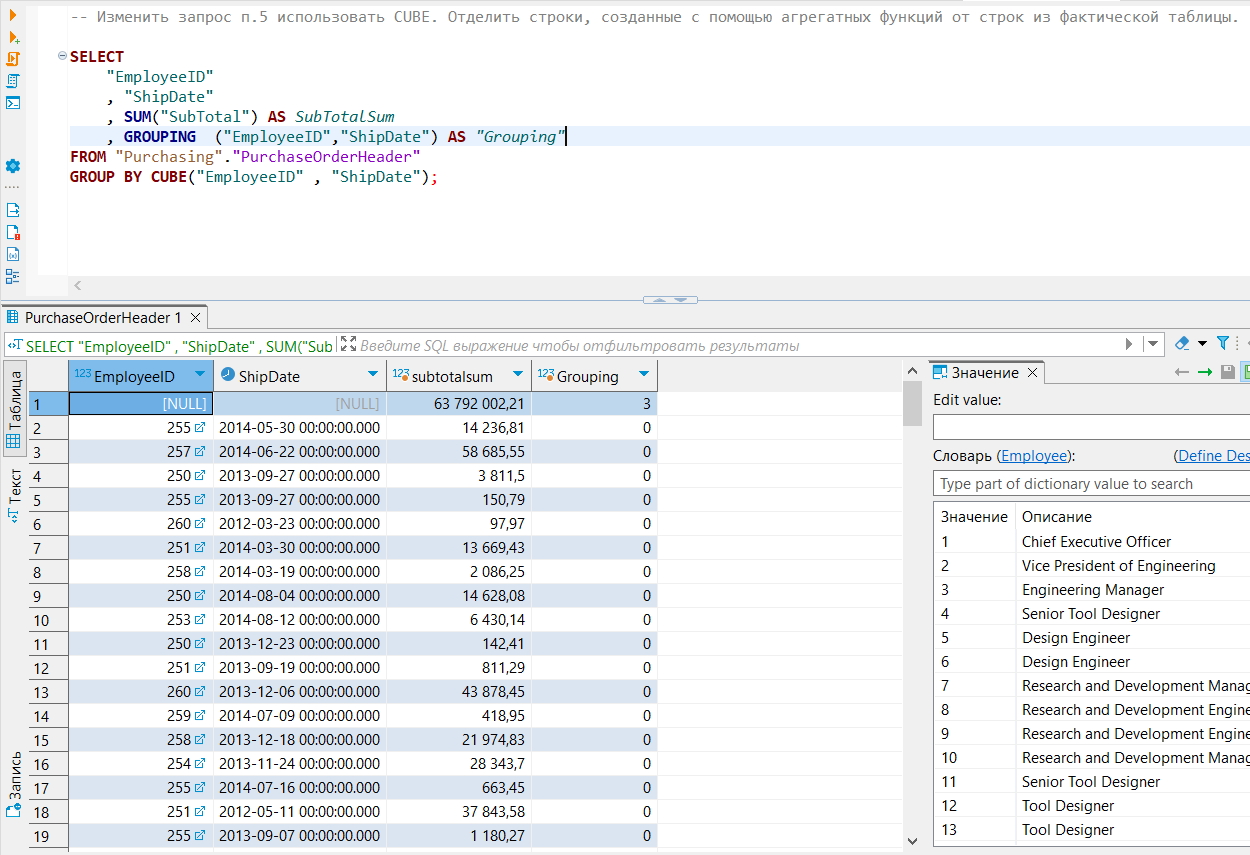


Рисунок 6 − Результат выполнения шестого задания

Задание 7

1. Формулировка задания:

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

1. Текст скрипта:

--Изменить запрос п.5 использовать GROUPING SETS. Отделить строки,

--созданные с помощью агрегатных функций от строк из фактической таблицы.

**SELECT**

"EmployeeID"

, "ShipDate"

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

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

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

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

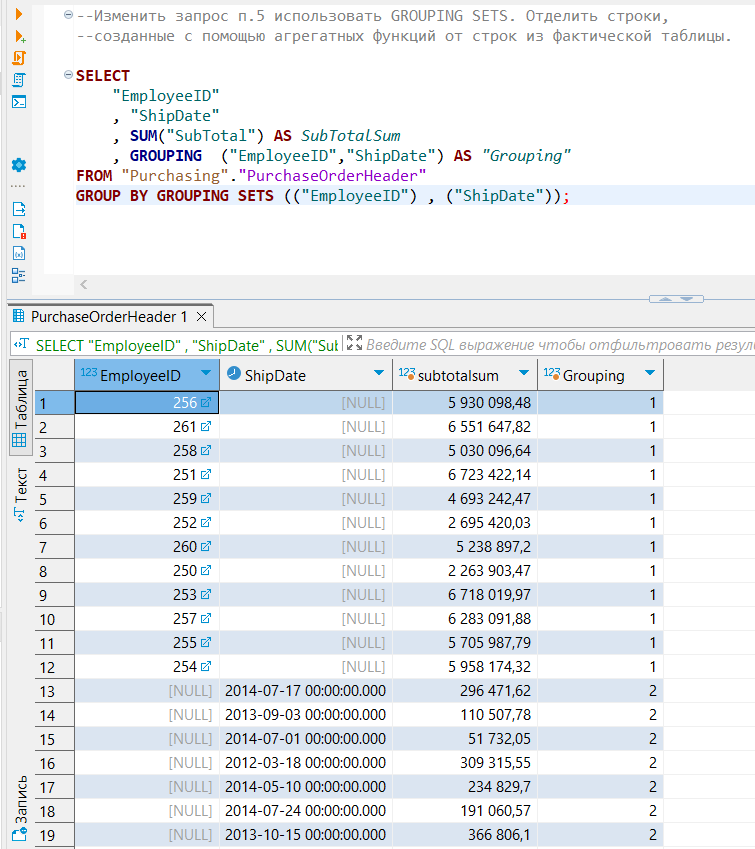


Рисунок 7 − Результат выполнения седьмого задания

## Отметка о выполнении лабораторной работы в веб-хостинге GitHub

Подтверждение о правильно выполненной работе представлено ниже (Рисунок 8).

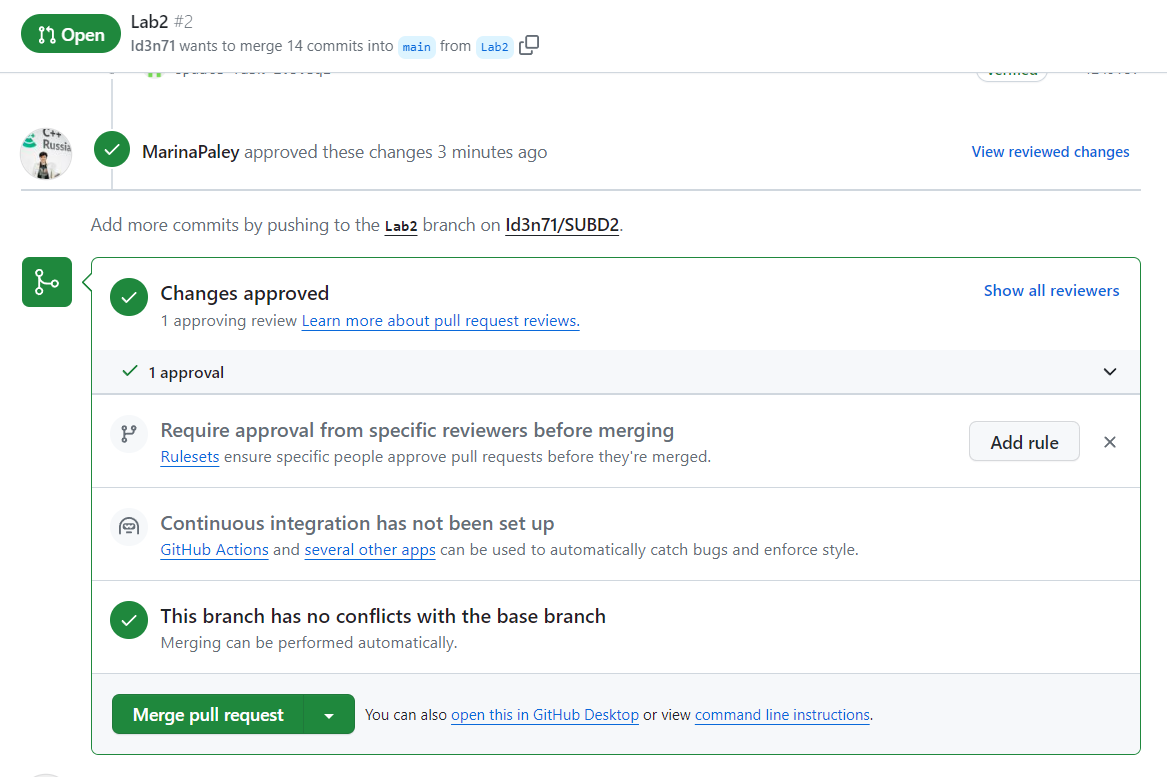


Рисунок 8 − Подтверждение о правильно выполненной работе