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 19

Выполнил: ст. гр. ТКИ-441

Чекан Фёдор Сергеевич

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. Формулировка задания:

Найти средний вес из таблицы Production.Product.

1. Текст скрипта:

--Найти средний вес из таблицы Production.Product

select AVG("Weight") as "AvgWeight"

from "Production"."Product";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

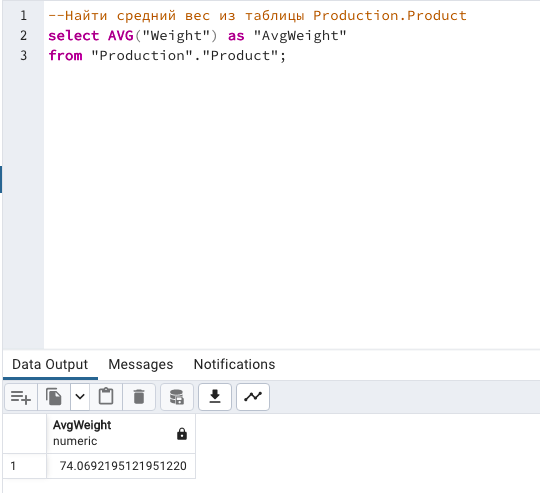


Рисунок  − Результат выполнения первого задания

Задание

1. Формулировка задания:

Найти количество людей, которых зовут Barry Johnson, из таблицы Person.Person.

1. Текст скрипта:

--Найти количество людей, которых зовут Barry Johnson, из

--таблицы Person.Person

select COUNT(\*) as "CountBarryJohnson"

from "Person"."Person"

where "FirstName" = 'Barry'

and "LastName" = 'Johnson';

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

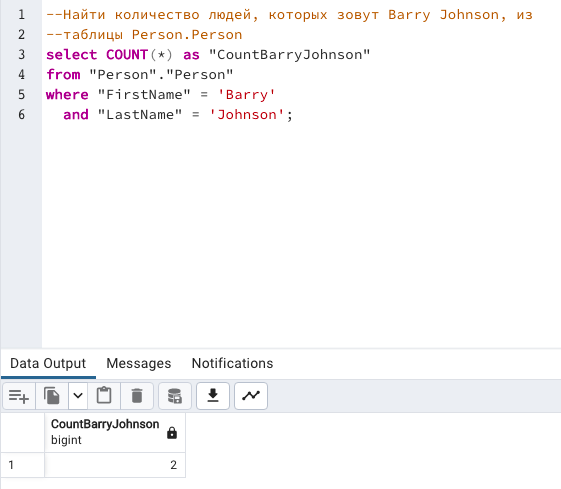


Рисунок  − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Показать суммарную стоимость товаров в каждой линейке продуктов ProductLine из таблицы Production.Product (из выборки исключить пустые поля).

1. Текст скрипта:

--Показать суммарную стоимость товаров в каждой линейке

--продуктов ProductLine из таблицы Production.Product (из

--выборки исключить пустые поля).

select "ProductLine", SUM("ListPrice") as "SumPrice"

from "Production"."Product"

where "ProductLine" is not null

group by "ProductLine";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок ).

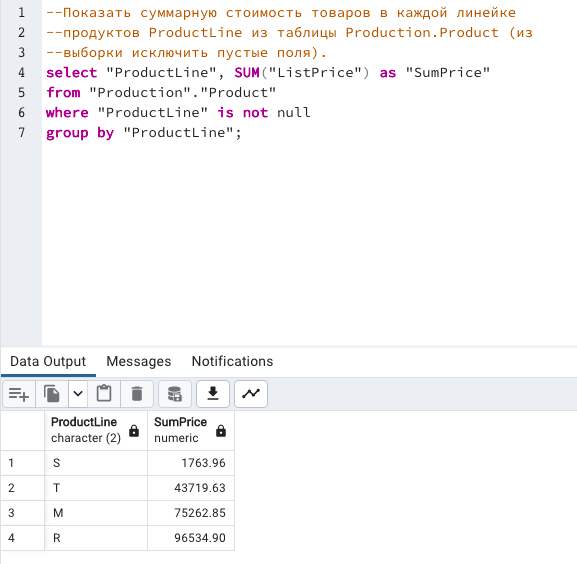


Рисунок  − Результат выполнения третьего задания

Задание 4

1. Формулировка задания:

Найти профессии (JobTitle) из таблицы HumanResources.Employee, где количество работников мужского пола (Gender) больше 2 (из выборки исключить пустые поля).

1. Текст скрипта:

--Найти профессии (JobTitle) из таблицы

--HumanResources.Employee, где количество работников

--мужского пола (Gender) больше 2 (из выборки исключить пустые

--поля).

select "JobTitle"

from "HumanResources"."Employee"

where "Gender" = 'M'

and "JobTitle" is not null

GROUP BY "JobTitle"

HAVING COUNT(\*) > 2;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

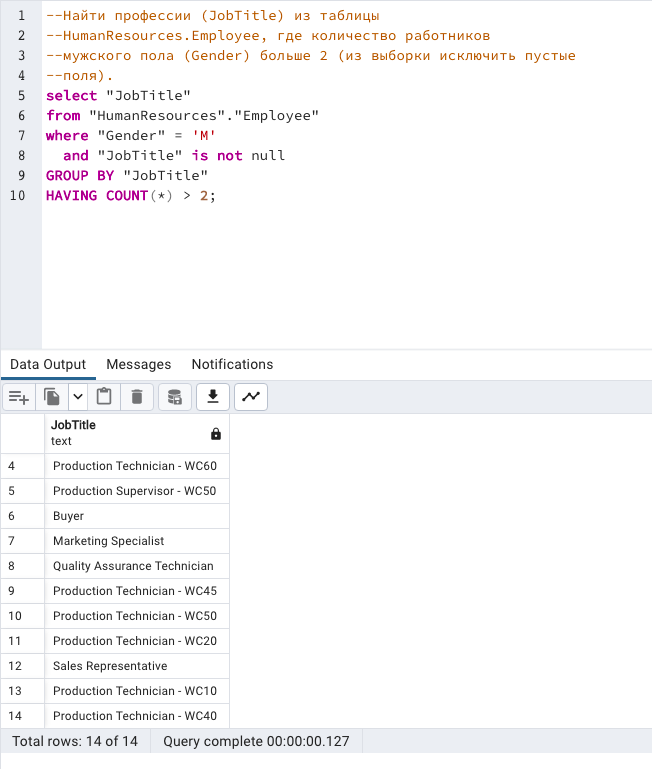


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

1. Формулировка задания:

Вывести TerritoryID, StoreID и количество StoreID, предусмотреть вывод количества StoreID для всех StoreID имеющих значение (NOT NULL) из таблицы из таблицы Sales.Customer. (Использовать ROOLUP).

1. Текст скрипта:

--Вывести TerritoryID, StoreID и количество StoreID,

--предусмотреть вывод количества StoreID для всех StoreID

--имеющих значение (NOT NULL) из таблицы из таблицы

--Sales.Customer. (Использовать ROOLUP).

select "TerritoryID", "StoreID", COUNT("StoreID")

from "Sales"."Customer"

GROUP BY ROLLUP ("TerritoryID", "StoreID")

having "StoreID" IS NOT NULL;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

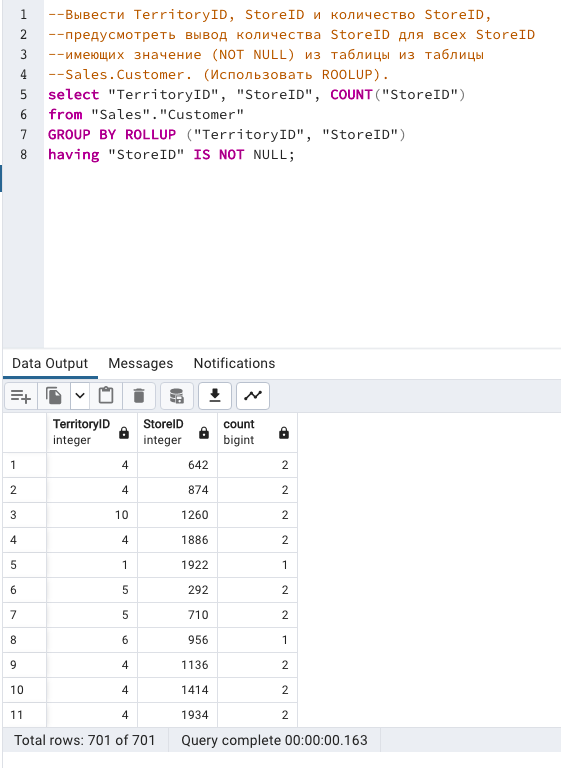


Рисунок 5 − Результат выполнения пятого задания

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 "TerritoryID", "StoreID", COUNT("StoreID"), GROUPING("TerritoryID", "StoreID")

from "Sales"."Customer"

GROUP BY CUBE ("TerritoryID", "StoreID");

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

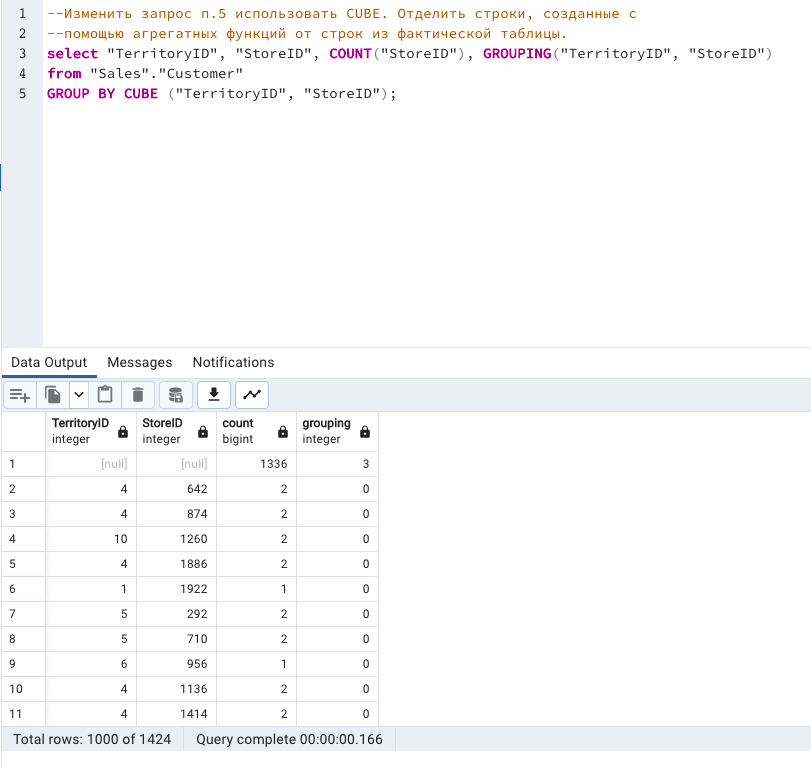


Рисунок 6 − Результат выполнения шестого задания

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 "TerritoryID", "StoreID", COUNT("StoreID"), GROUPING("TerritoryID", "StoreID")

from "Sales"."Customer"

GROUP BY GROUPING SETS (("TerritoryID"), ("StoreID"), ());

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

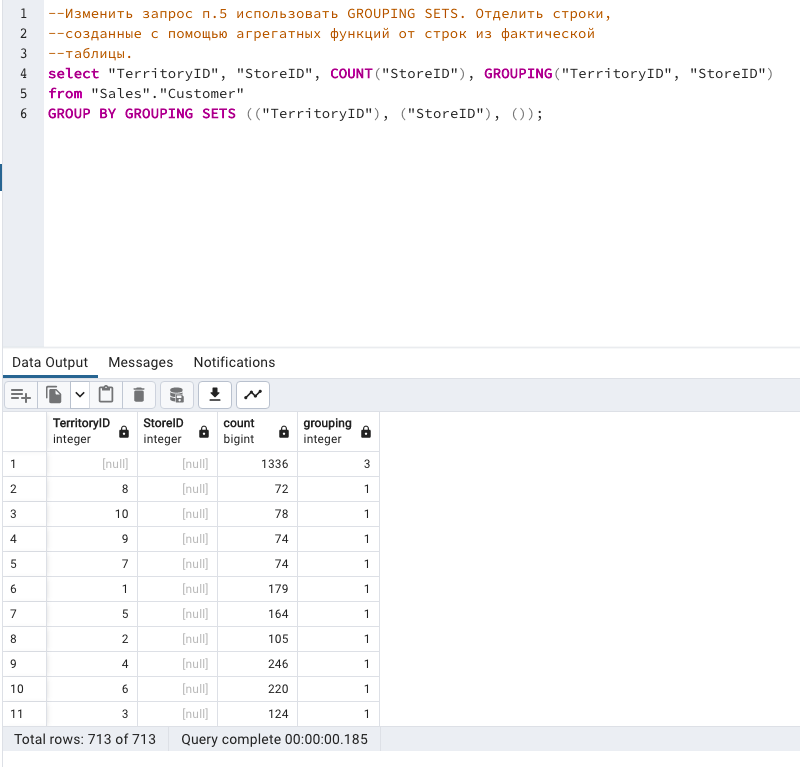


Рисунок 7 − Результат выполнения седьмого задания

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

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

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 8 − Подтверждение о правильно выполненной работе