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 13

Выполнил: ст. гр. ТКИ-441

Фамилия Порхун Д.Д.

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

Москва 2024

## Цель работы

Выполнить вторую лабораторную работу, Изучить операторы SQL диалекта PostgresPro, необходимые для группировки и обобщения данных. Научиться создавать запросы с агрегатными[[1]](#footnote-1) функциями. Подготовить отчет, выполненный согласно требованиям ГОСТ 7.32−2017.

## РЕЗУЛЬТАТЫ ВЫПОЛНЕННОЙ РАБОТЫ

Задание 1

1. Формулировка задания:

Найти среднюю стоимость (StandardCost) из таблицы ProductCostHistory.

1. Текст скрипта:

**select** **avg**("StandardCost") **from** "Production"."ProductCostHistory";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).



Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

Найти пароли, начинающиеся с I или i, из таблицы Person.Password.

1. Текст скрипта:

**select** "PasswordSalt" **from** "Person"."Password" **where** "PasswordSalt" **LIKE** **'I%'** **or** "PasswordSalt" **LIKE** **'i%'**

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).



Рисунок 2 − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Найти сумму часов выходных для разных полов из таблицы HumanResources.Employee.

1. Текст скрипта:

**SELECT** **SUM**(**CASE** **WHEN** "Gender" = **'M'** **THEN** "SickLeaveHours" **ELSE** 0 **END**) **AS** *"SumTimeMale"*, **SUM**(**CASE** **WHEN** "Gender" = **'F'** **THEN** "SickLeaveHours" **ELSE** 0 **END**) **AS** *"SumTimeFemale"*

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

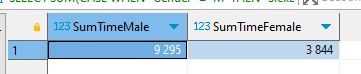


Рисунок 3 − Результат выполнения второго задания

Задание 4

1. Формулировка задания:

Показать города, основываюсь на таблице Person.Address, где количество заказов было больше 10 (из выборки исключить пустые поля).

1. Текст скрипта:

**SELECT** "City" , **COUNT**("StateProvinceID") **FROM** "Person"."Address"

**WHERE** "StateProvinceID" **IS** **NOT** **NULL**

**GROUP** **by** "City"

**having** **COUNT**("StateProvinceID") > 10;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

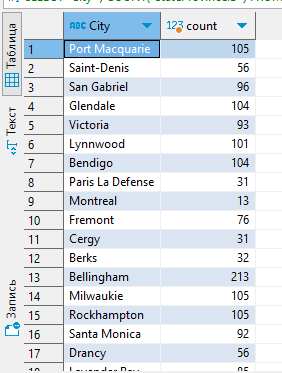


Рисунок 4 − Результат выполнения второго задания

Задание 5

1. Формулировка задания:

Вывести ProductID из таблицы Production.ProductListPriceHistory у которых максимальная цена (ListPrice) больше 24. Добавить столбец с рангом, определяя его порядок в зависимости от количества строк, содержащих максимальную цену. (Использовать RANK).

1. Текст скрипта:

**select** "ProductID", **RANK**() **OVER**(**ORDER** **BY** **count**("ProductID") **DESC**), **count**("ProductID") **as** *"Count"*

**from** "Production"."ProductListPriceHistory" **where** "ListPrice" > 24 **group** **by** "ProductID";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

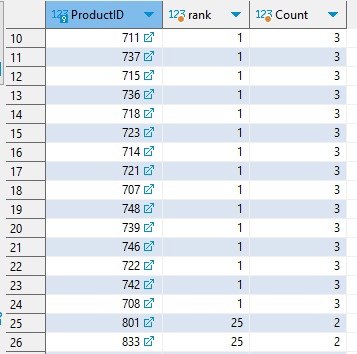


Рисунок 5 − Результат выполнения второго задания

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

Подтверждение о правильно выполненной работе представлено ниже (Рисунок 6).

Тут будет скрин апрува

Рисунок 6 − Подтверждение о правильно выполненной работе

1. статистические функции [↑](#footnote-ref-1)