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 №4

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 1, необходимые для работы с подзапросами. Научиться создавать вложенные и коррелированные запросы. Подготовить отчет, выполненный согласно требованиям ГОСТ 7.32−2017.

## РЕЗУЛЬТАТЫ ВЫПОЛНЕННОЙ РАБОТЫ

Задание

1. Формулировка задания:

Показать товары, цена которых ниже средней цены (не учитывать товары с ценой, равной нулю) (Таблица Production.Product).Вывести Name и ListPrice.

1. Текст скрипта:

--Показать товары, цена которых ниже средней цены (не учитывать

--товары с ценой, равной нулю) (Таблица Production.Product). Вывести Name и ListPrice.

**SELECT** "Name" , "ListPrice"

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

**WHERE** "ListPrice" > 0

**AND** "ListPrice" < (

**SELECT** **AVG**("ListPrice")

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

**WHERE** "ListPrice" > 0

);

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

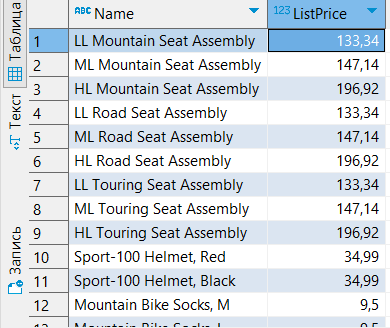


Рисунок  − Результат выполнения первого задания

Задание

1. Формулировка задания:

Ранжирование товаров по размеру: от меньшего к большему. ("Production"."Product"). Примечание к ранжированию – ранг может начинаться с произвольного числа (например, 2 или 5), главное – ранг не должен уменьшаться.

1. Текст скрипта:

--Ранжирование товаров по размеру: от меньшего к большему.

--("Production"."Product"). Примечание к ранжированию – ранг

--может начинаться с произвольного числа (например, 2 или 5), главное – ранг не должен уменьшаться.

**SELECT** "Name" , "ListPrice" ,

(**SELECT** **COUNT**(\*)

**FROM** "Production"."Product" **AS** *p2*

**WHERE** *p2*."ListPrice" < *p1*."ListPrice") **AS** *rank*

**FROM** "Production"."Product" **AS** *p1*

**ORDER** **BY** *rank*;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

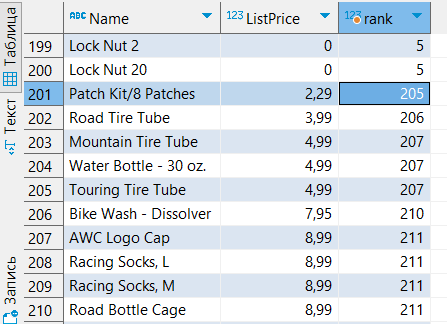


Рисунок  − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Получите список всех PurchaseOrderDetailID, у которых UnitPrice равна UnitPrice c PurchaseOrderID = 31

1. Текст скрипта:

--Получите список всех PurchaseOrderDetailID, у которых

--UnitPrice равна UnitPrice c PurchaseOrderID = 31

**SELECT** "PurchaseOrderDetailID"

**FROM** "Purchasing"."PurchaseOrderDetail" *pod2*

**WHERE** "UnitPrice" = (

**SELECT** "UnitPrice"

**FROM** "Purchasing"."PurchaseOrderDetail" *pod*

**WHERE** "PurchaseOrderID" = 31

**LIMIT** 1

);

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок ).

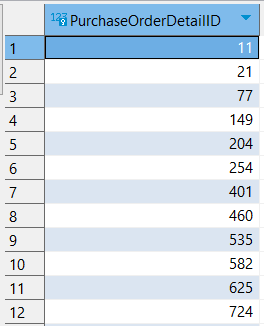


Рисунок  − Результат выполнения третьего задания

Задание 4

1. Формулировка задания:

Вывести имя и фамилию сотрудника, а также фамилию следующего в алфавитном порядке сотрудника. Учитывать только сотрудников, получивших зарплату. ("Person"."Person", "HumanResources"."Employee").

1. Текст скрипта:

--Вывести имя и фамилию сотрудника, а также фамилию следующего в

--алфавитном порядке сотрудника. Учитывать только сотрудников,получивших зарплату.("Person"."Person","HumanResources"."Employee").

**SELECT** *p1*."FirstName" , *p1*."LastName" ,

(**SELECT** *p2*."LastName"

**FROM** "HumanResources"."Employee" *e2*

**JOIN** "Person"."Person" *p2* **ON** *e2*."BusinessEntityID" = *p2*."BusinessEntityID"

**WHERE** *e2*."SalariedFlag" = 1

**AND** *p2*."LastName" > *p1*."LastName"

**ORDER** **BY** *p2*."LastName"

**LIMIT** 1) **AS** *NextLastName*

**FROM** "HumanResources"."Employee" *e1*

**JOIN** "Person"."Person" *p1* **ON** *e1*."BusinessEntityID" = *p1*."BusinessEntityID"

**WHERE** *e1*."SalariedFlag" = 1

**ORDER** **BY** *p1*."LastName" ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

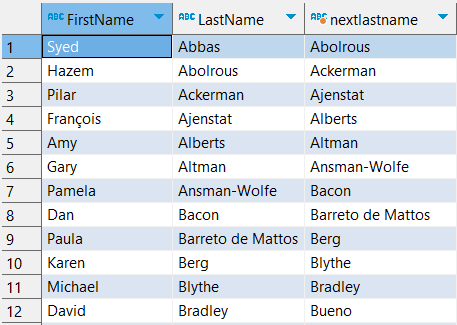


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

1. Формулировка задания:

Для каждого Product вывести отклонение от среднего SafetyStockLevel. Таблица "Production"."Product".

1. Текст скрипта:

--Для каждого Product вывести отклонение от среднего SafetyStockLevel. Таблица "Production"."Product".

**SELECT** "Name" , "SafetyStockLevel" ,

"SafetyStockLevel" - (

**SELECT** **AVG**("SafetyStockLevel")

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

) **AS** *DeviationFromAverage*

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

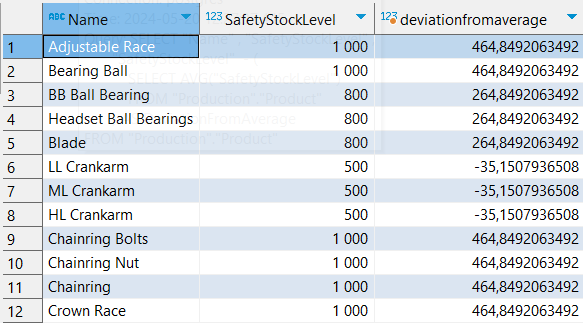


Рисунок 5 − Результат выполнения пятого задания

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

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

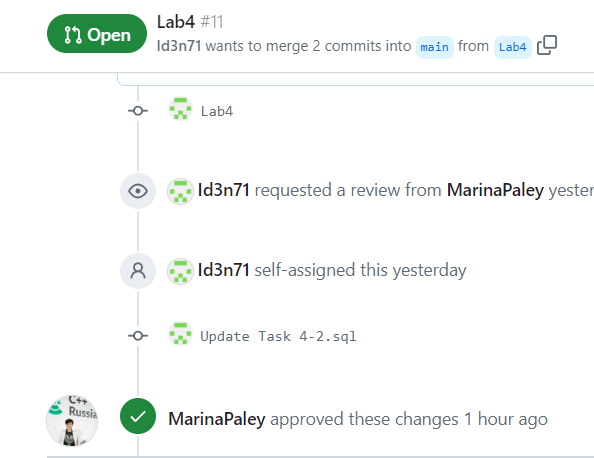


Рисунок 6 − Подтверждение о правильно выполненной работе