Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**О Т Ч Ё Т**

по лабораторной работе №4

по дисциплине:

«Информационное обеспечение систем управления»

на тему:

«Вложенные запросы»

Выполнил: ст. гр. ВУЦ-421

Полунин С.К.

Вариант 6

Проверил: доц., к.т.н.

Васильева М. А.

Москва ­2024

**Цель работы**

Изучить операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для работы с подзапросами. Научиться создавать вложенные и коррелированные запросы. Подготавливать отчет, выполненный согласно требованиям ГОСТ 7.32−2017.

**Результаты выполненной работы**

**Задание** **1**

Формулировка задания:

Показать товары, цена которых меньше средней цены (Таблица Production.Product). Показать поля Name, Size и ListPrice.

Текст скрипта:

**select**

*p*."Name"

,*p*."Size"

,*p*."ListPrice"

**from**

"Production"."Product" *p*

**where**

*p*."ListPrice" <

(

**select** **avg** ("ListPrice") **as** *"AvgListPrice"*

**from** "Production"."Product" *p2*

**where** "ListPrice" > 0

)

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

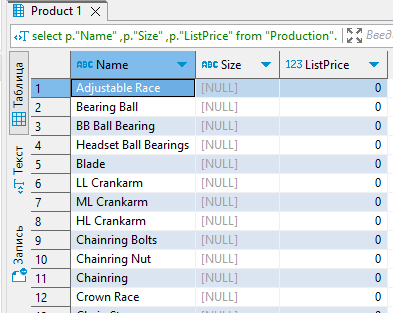


Рисунок 1 – Результат выполнения первого задания.

**Задание** **2**

Формулировка задания:

Ранжирование сотрудников по должности в алфавитном порядке. ("HumanResources"."Employee"). Примечание к ранжированию – ранг может начинаться с произвольного числа (например, 2 или 5), главное – ранг не должен уменьшаться.

Текст скрипта:

**select** *e1*."JobTitle",

(**select** **count**(\*)

**from** "HumanResources"."Employee" *e2*

**where** *e2*."JobTitle" <= *e1*."JobTitle") **as** *Rank*

**from** "HumanResources"."Employee" *e1*

**order** **by** *e1*."JobTitle"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).



Рисунок 2 – Результат выполнения второго задания.

**Задание** **3**

Формулировка задания:

Получите список всех PurchaseOrderDetailID, у которых UnitPrice выше UnitPrice c PurchaseOrderID = 73

Текст скрипта:

**select** *pod*."PurchaseOrderDetailID"

**from** "Purchasing"."PurchaseOrderDetail" *pod*

**where** *pod*."UnitPrice" >

(

**select** "UnitPrice"

**from** "Purchasing"."PurchaseOrderDetail"

**where** "PurchaseOrderID" = 73

)

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

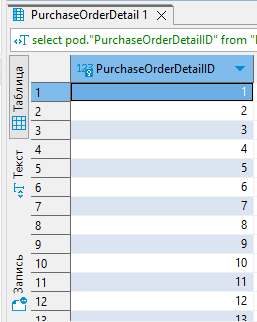


Рисунок 3 – Результат выполнения третьего задания.

**Задание** **4**

Формулировка задания:

Вывести название должности сотрудника, а также название должности сотрудника, имеющего большее количество отпускных часов. Учитывать только действующих сотрудников. ("HumanResources"."Employee").

Текст скрипта:

**select** *e1*."JobTitle",

(**select** *e2*."JobTitle"

**from** "HumanResources"."Employee" *e2*

**where** *e2*."VacationHours" =

(

**select** **max**(*e3*."VacationHours")

**from** "HumanResources"."Employee" *e3*

**where** *e3*."CurrentFlag" = 1

)

**and** *e2*."CurrentFlag" = 1

**limit** 1) **as** *"MaxHoursJobTitle"*

**from** "HumanResources"."Employee" *e1*

**where** *e1*."CurrentFlag" = 1

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

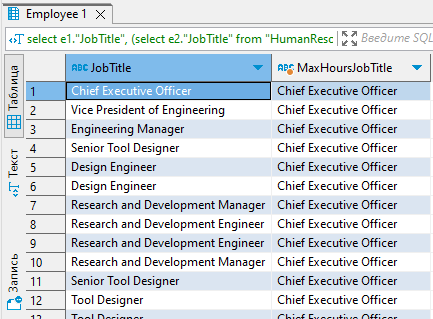


Рисунок 4 − Результат выполнения четвертого задания.

**Задание** **5**

Формулировка задания:

Показать товары, цена которых меньше средней цены в том же стиле (Таблица Production.Product). Показать поля Name, ListPrice и Style.

Текст скрипта:

**select**

*p*."Name",

*p*."ListPrice",

*p*."Style"

**from** "Production"."Product" *p*

**where** *p*."ListPrice" <

(

**select** **avg**(*p2*."ListPrice")

**from** "Production"."Product" *p2*

**where** *p2*."Style" = *p*."Style"

)

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

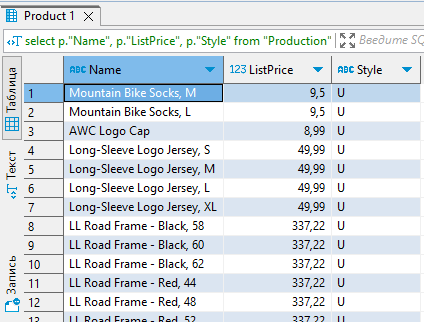


Рисунок 5 − Результат выполнения пятого задания.

Вывод: изучены операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для работы с подзапросами, а так же научились создавать вложенные и коррелированные запросы.