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 №1

По дисциплине «Информационное обеспечение систем управления»

ВАРИАНТ 5

Выполнил: ст. гр. ВУЦ - 451

Лихачев Сергей Святославович

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

Москва 2024

## Цель работы

Изучить операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для фильтрации набора данных. Научиться создавать простые запросы на фильтрацию данных. Подготовить отчет, выполненный согласно требованиям ГОСТ 7.32−2017.

## РЕЗУЛЬТАТЫ ВЫПОЛНЕННОЙ РАБОТЫ

Задание 1

Формулировка задания:

Показать все поля из таблицы Production.WorkOrderRouting.

Текст скрипта:

**select** \* **from** "Production"."WorkOrderRouting" wor

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

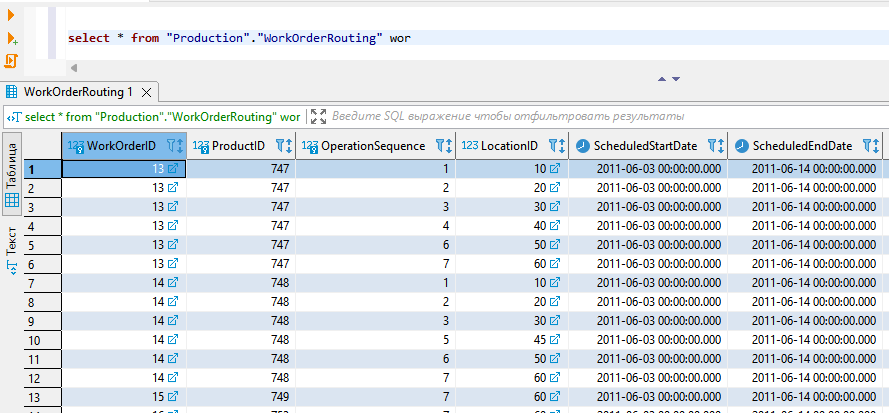


Рисунок 1 – Результат выполнения первого задания

Задание 2

Формулировка задания:

Показать поля ProductDescriptionID, [Description] из таблицы Production.ProductModelProductDescription.

Текст скрипта:

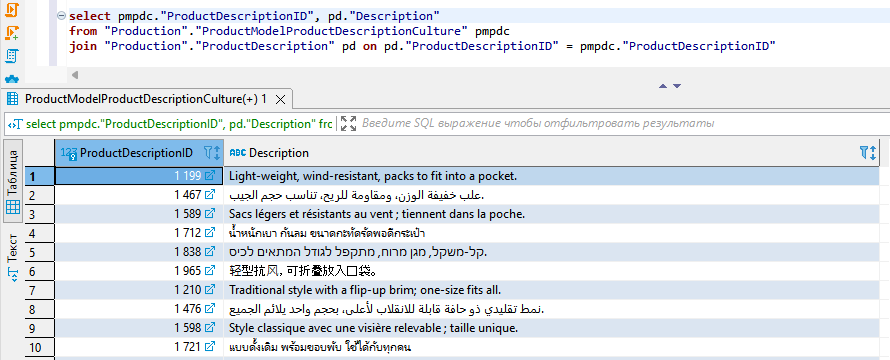
**select** pmpdc."ProductDescriptionID", pd."Description"

**from** "Production"."ProductModelProductDescriptionCulture" pmpdc

**join** "Production"."ProductDescription" pd **on** pd."ProductDescriptionID" = pmpdc."ProductDescriptionID"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

**Рисунок 2 – Результат выполнения второго задания**

Задание 3

Формулировка задания:

Показать поля BusinessEntityID, RateChangeDate, Rate из таблицы HumanResources.EmployeePayHistory. Показать только те записи, для которых почасовая ставка заработной платы (Rate) больше $15 и меньше $20. Использовать оператор BETWEEN.

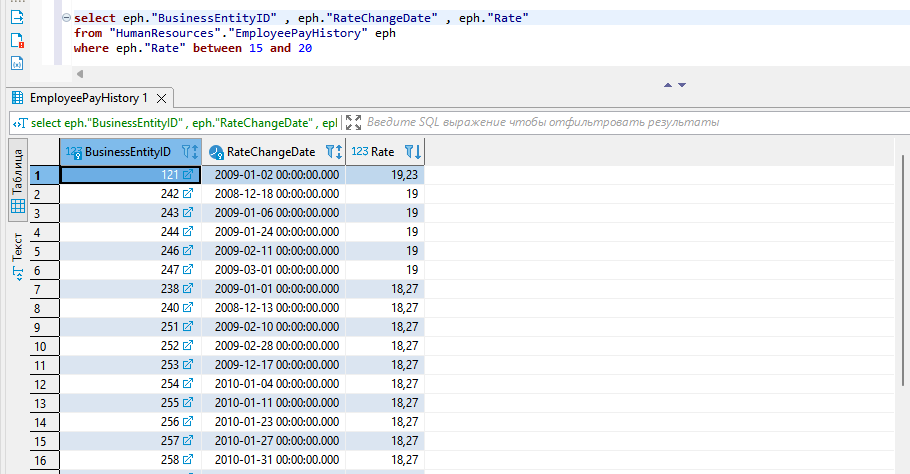
Текст скрипта:

**select** eph."BusinessEntityID" , eph."RateChangeDate" , eph."Rate"

**from** "HumanResources"."EmployeePayHistory" eph

**where** eph."Rate" **between** 15 **and** 20

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

**Рисунок 3 – Результат выполнения третьего задания**

Задание 4

Формулировка задания:

Из таблицы Sales.SalesOrderDetail показать ID продажи (SalesOrderID) и ID товара (ProductID). Показать только те товары, у которых есть CarrierTrackingNumber.

Текст скрипта:

**select** sod."SalesOrderID" , sod."ProductID"

**from** "Sales"."SalesOrderDetail" sod

**where** sod."CarrierTrackingNumber" **is** **not** **null**

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

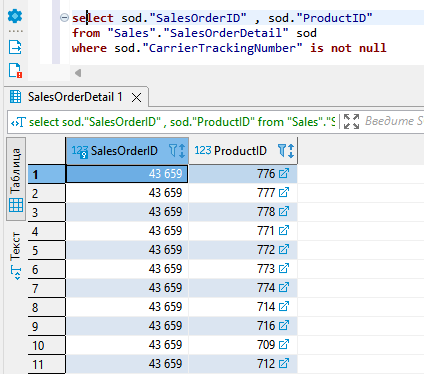


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

Формулировка задания:

Показать поля ProductCategoryID и Name из таблицы Production.ProductCategory. Показать только те товары, название которых начинается на букву 'C'.

Текст скрипта:

**select** pc."ProductCategoryID" , pc."Name"

**from** "Production"."ProductCategory" pc

**where** pc."Name" **like** 'C%'

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

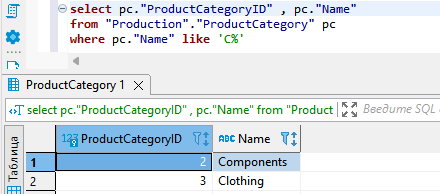


Рисунок 5 − Результат выполнения пятого задания

Задание 6

Формулировка задания:

Из таблицы Sales.Store показать ID сотрудника (BusinessEntityID), название магазина (Name) и ID продавца (SalesPersonID). Показать только значения поля Name, принадлежащие списку ('Next-Door Bike Store', 'Area Bike Accessories', 'Top of the Line Bikes', 'Valley Toy Store', 'Global Plaza'). Использовать оператор IN.

Текст скрипта:

**select** s."BusinessEntityID" , s."Name" , s."SalesPersonID"

**from** "Sales"."Store" s

**where** s."Name" **in**

('Next-Door Bike Store',

'Area Bike Accessories',

'Top of the Line Bikes',

'Valley Toy

Store',

'Global Plaza')

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

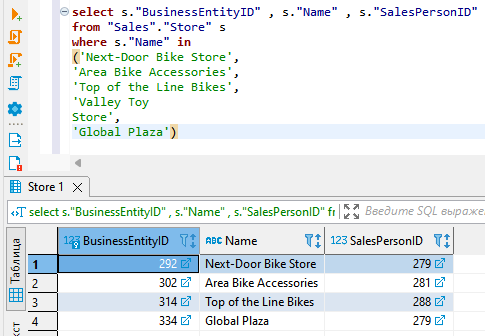


Рисунок 6 − Результат выполнения шестого задания

Задание 7

Формулировка задания:

Показать поля BusinessEntityID, AccountNumber, Name, PurchasingWebServiceURL из таблицы Purchasing.Vendor. Неизвестные адреса сайтов магазинов (PurchasingWebServiceURL) заменить значением 'not available'. Названия полей оставить без изменений.

Текст скрипта:

**select** v."BusinessEntityID" , v."AccountNumber" , v."Name" ,

**coalesce** (v."PurchasingWebServiceURL", 'not available') **as** "PurchasingWebServiceURL"

**from** "Purchasing"."Vendor" v

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

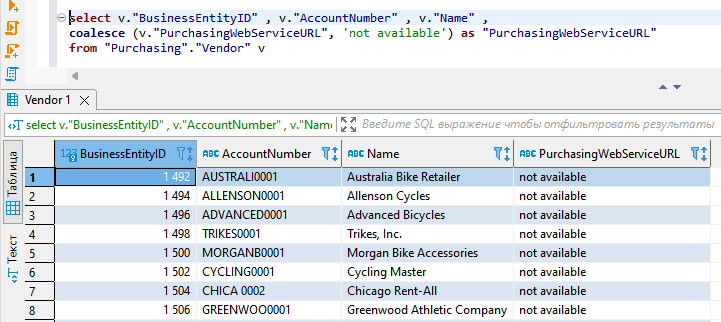


Рисунок 7 − Результат выполнения седьмого задания

Задание 8

Формулировка задания:

Показать поля ProductID, Name из таблицы Production.Product с применением функции COALESCE(). Показать поле Meauserement, так, чтобы, если значение в поле Style известно, то показать его, а иначе, показать значение в поле Color. Если и в поле Color значение неизвестно, то вывести значение 'UNKNOWN'.

Текст скрипта:

**select** p."ProductID" , p."Name" ,

**coalesce** ("Style", "Color", 'unlnown') **as** "Meauserement"

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

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

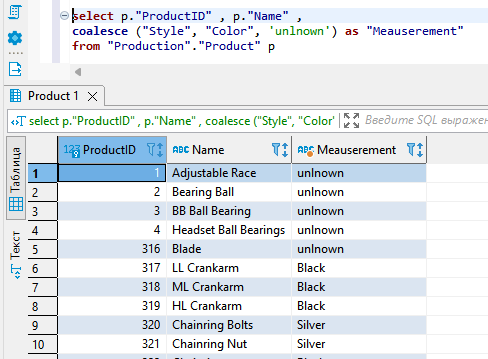


Рисунок 8 − Результат выполнения восьмого задания

Задание 9

Формулировка задания:

Из таблицы Production.Culture показать значения поля Name в алфавитном порядке.

Текст скрипта:

**select** c."Name"

**from** "Production"."Culture" c

**order** **by** c."Name"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).

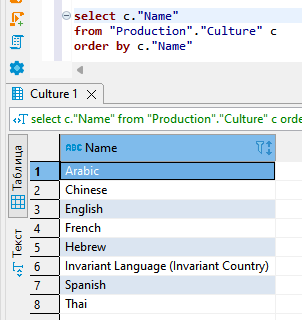


Рисунок 9 − Результат выполнения девятого задания

Задание 10

Формулировка задания:

Показать название (Name) и размер (Size) товара из таблицы Production.Product, если имеются товары только размера 'M'. Заменить размер товара на NULL, если он имеет значение 'M'. Названия поля Size заменить на Class.

Текст скрипта:

**select** p."Name" ,

**case**

**when** p."Size" = 'M'**then** **null**

**else** p."Size"

**end** **as** "Class"

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

**where** "Size" ='M'

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10).

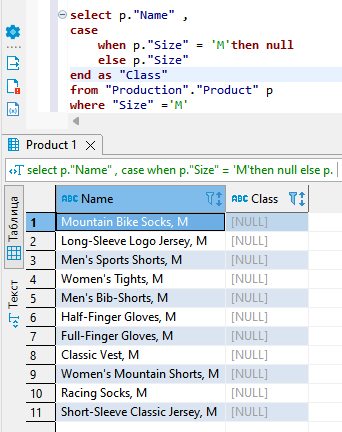


Рисунок 10 − Результат выполнения десятого задания

## Вывод

В результате выполнения лабораторной работы были изучены операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для фильтрации набора данных. Получен навык создавать простые запросы на фильтрацию данных.