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 «Основы построения защищенных баз данных»

ВАРИАНТ 10

Выполнил: ст. гр. ТКИ-442

Пшеницын Пётр Евгеньевич

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

1. Формулировка задания:

Показать все поля из таблицы Production.Product.

1. Текст скрипта:

SELECT \* FROM "Production"."Product";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

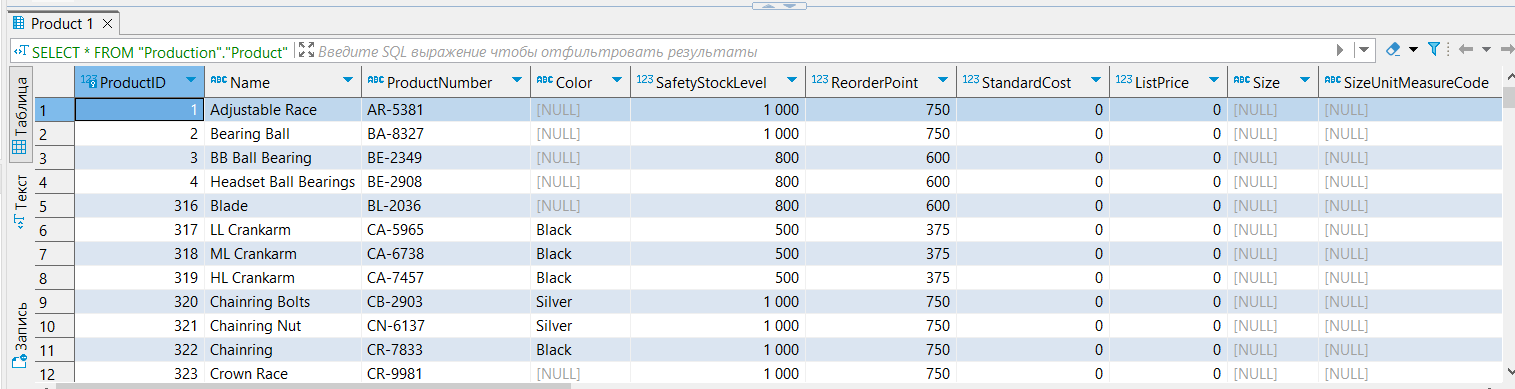


Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

Показать поля FirstName, LastName и MiddleName из Person.Person Текст скрипта:

SELECT "FirstName" , "LastName" ,"MiddleName"

FROM "Person"."Person" ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

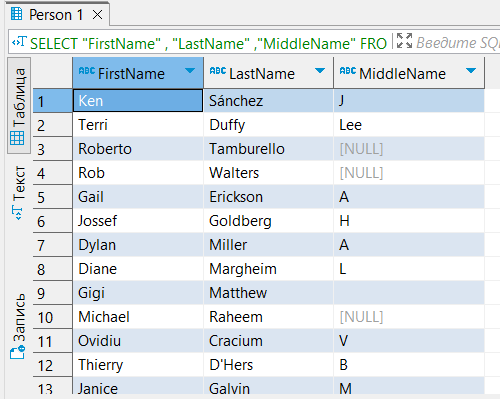


Рисунок 2 − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Показать поля ProductId, UnitPrice и OrderQty из таблицы Purchasing.PurchaseOrderDetail, где количество заказанного товара (OrderQty) больше 60, а цена (UnitPrice) находится в диапазоне от $50 до $100.

1. Текст скрипта:

SELECT "ProductID"

,"UnitPrice"

, "OrderQty"

FROM "Purchasing"."PurchaseOrderDetail"

where ("OrderQty" >= '60' or "UnitPrice" >= '50' or "UnitPrice" <= '100')

;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

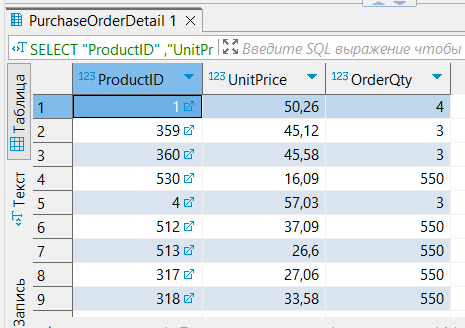


Рисунок 3 − Результат выполнения первого задания

Задание 4

1. Формулировка задания:

Показать идентификационный номер сотрудника (BusinessEntityID), номер аккаунта (AccountNumber), название поставщика (Name) и URL поставщика (PurchasingWebServiceURL) из таблицы Purchasing.Vendor, где у поставщика имеется URL .

1. Текст скрипта:

SELECT "BusinessEntityID"

,"AccountNumber" ,

"Name"

,"PurchasingWebServiceURL"

FROM "Purchasing"."Vendor" ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

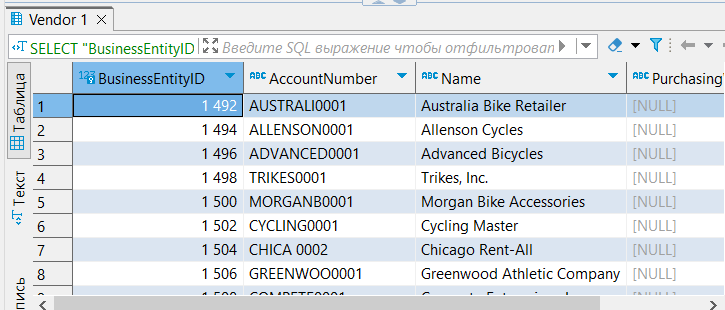


Рисунок 4 − Результат выполнения первого задания

Задание 5

1. Формулировка задания:

Показать все названия товаров (Name), содержащие 'Frame' («рама») и ID моделей товаров (ProductModelID) из таблицы Production.ProductModel.

1. Текст скрипта:

SELECT "Name" ,"ProductModelID"

FROM "Production"."ProductModel"

WHERE "Name" LIKE '%Frame%';

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

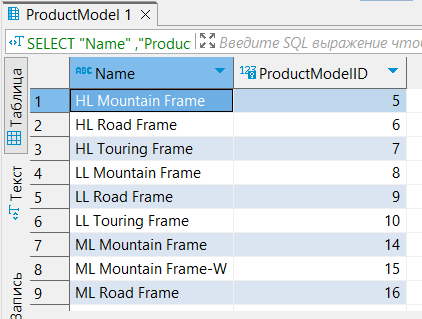


Рисунок 5 − Результат выполнения первого задания

Задание 6

1. Формулировка задания:

Показать сотовый и рабочий телефонный номер (PhoneNumber), идентификационный номер сотрудника (BusinessEntityID) и вид телефонного номера (PhoneNumberTypeID) из таблицы Person.PersonPhone, где вид телефонного номера или 1 (сотовый), или 3 (рабочий). Использовать оператор IN. Отсортировать строки в алфавитном порядке относительно поля BusinessEntityID .

1. Текст скрипта:

SELECT "PhoneNumber" ,"BusinessEntityID" , "PhoneNumberTypeID"

FROM "Person"."PersonPhone"

WHERE "PhoneNumberTypeID" IN ( 1, 3) ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

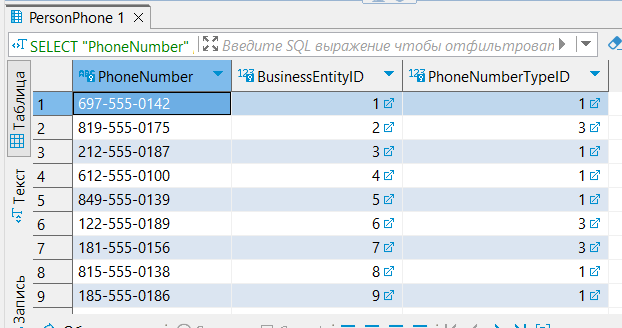


Рисунок 6 − Результат выполнения первого задания

Задание 7

1. Формулировка задания:

Показать поля AddressID, AddressLine1, AddressLine2, City из таблицы Person.[Address]. Заменить неизвестное значение в поле AddressLine2 на 'NO'. Названия полей оставить без изменений.

1. Текст скрипта:

SELECT "AddressID" ,"AddressLine1" ,"AddressLine2" ,"City",

COALESCE ("AddressLine2", 'NO') as "AddressLine2"

FROM "Person"."Address" ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

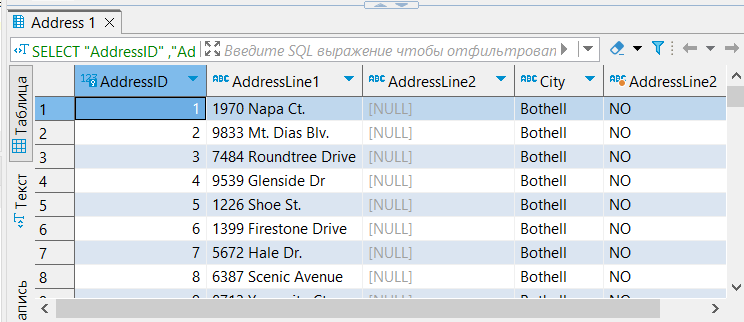


Рисунок 7 − Результат выполнения первого задания

Задание 8

1. Формулировка задания:

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

1. Текст скрипта:

SELECT "ProductID" ,"Name" ,

COALESCE ("Size","Color" ,'UNKNOWN' ) as Measurement

FROM "Production"."Product" ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

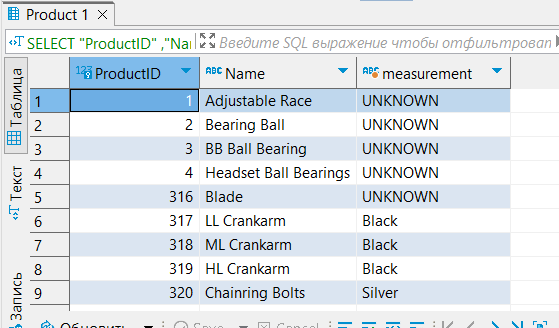


Рисунок 8 − Результат выполнения первого задания

Задание 9

1. Формулировка задания:

Отсортировать строки таблицы Production.ScrapReason в обратном алфавитном порядке значения брака Name. Показать поля ScrapReasonID, Name, ModifiedDate.

1. Текст скрипта:

SELECT "ScrapReasonID" ,"Name" ,"ModifiedDate"

FROM "Production"."ScrapReason"

ORDER BY "Name" DESC ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).

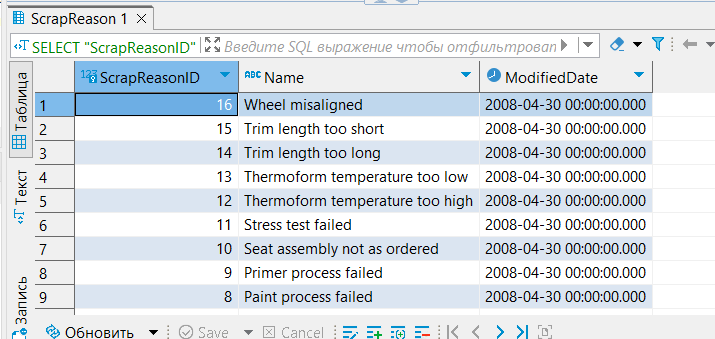


Рисунок 9 − Результат выполнения первого задания

Задание 10

1. Формулировка задания:

Заменить на NULL значение в поле ListPrice, если цена равна нулю. Показать поля ProductID, Name, Size, Color из таблицы Production.Product..

1. Текст скрипта:

SELECT "ProductID" ,"Name" ,"Size" ,"Color" ,

NULLIF ("Size" , '0') AS "Size"

FROM "Production"."Product"

WHERE "Size" IS NOT NULL;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10 ).

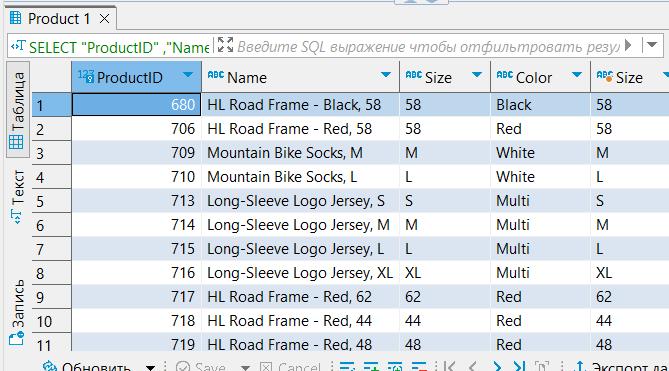


Рисунок 10 − Результат выполнения первого задания

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

Подтверждение о правильно выполненной работе представлено ниже (Рисунок 11).

--------------------------------------------------------------------

Рисунок 11 − Подтверждение о правильно выполненной работе