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 5

Выполнил: ст. гр. ТКИ-442

Зинченко Б.А.

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.WorkOrderRouting.

1. Текст скрипта:

select \*

from "Production"."WorkOrderRouting" wor ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

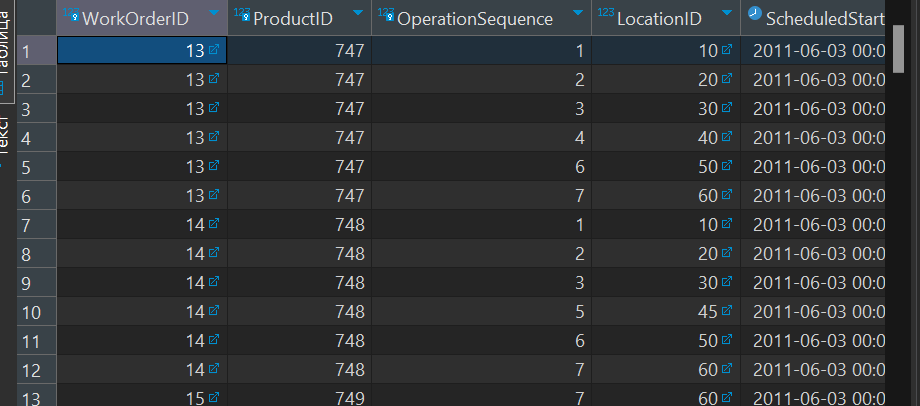


Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

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

1. Текст скрипта:

select

"TransactionID",

"ProductID",

"ReferenceOrderID",

"TransactionType",

"Quantity",

"ActualCost"

from "Production"."TransactionHistory";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

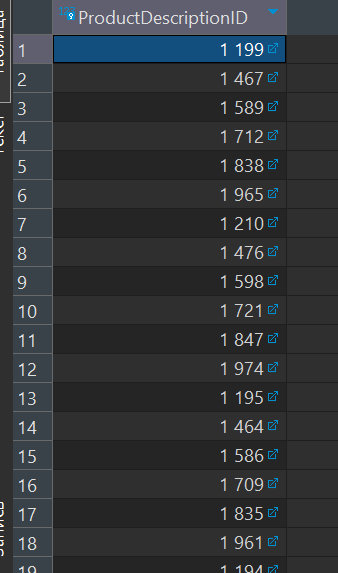


Рисунок 2 − Результат выполнения второго задания

Задание 3

1. 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.

1. Текст скрипта:

select

"BusinessEntityID"

,"RateChangeDate"

,"Rate"

from "HumanResources"."EmployeePayHistory" eph

where "Rate" between 15 and 20;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

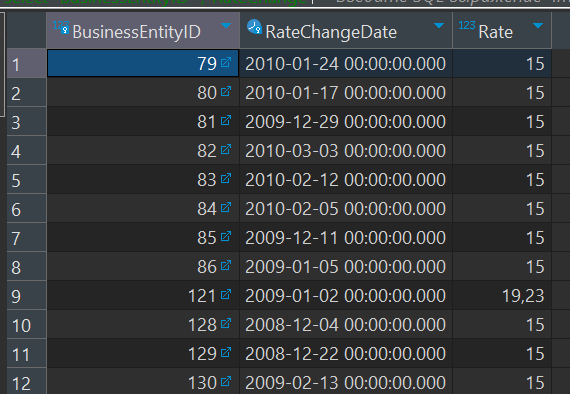


Рисунок 3 − Результат выполнения третьего-четвертого задания

Задание 4

1. Формулировка задания:

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

1. Текст скрипта:

select "SalesOrderID",

"ProductID"

from "Sales"."SalesOrderDetail"

where "CarrierTrackingNumber" is not null;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

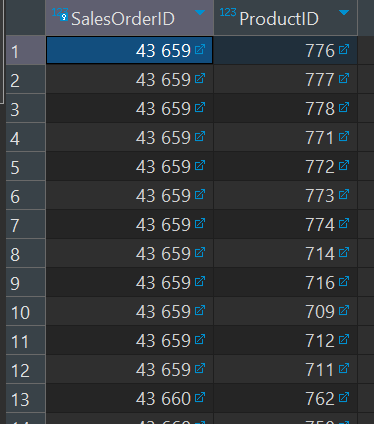


Рисунок 4 − Результат выполнения пятого задания

Задание 5

1. Формулировка задания:

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

1. Текст скрипта:

select

"ScrapReasonID",

"Name"

from "Production"."ScrapReason"

where "Name" in ('Color incorrect','Drill size too large','Drill size too small','Paint process failed');

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

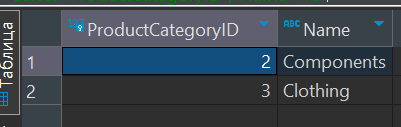


Рисунок 5 − Результат выполнения шестого задания

Задание 6

1. 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 To Store', 'Global Plaza'). Использовать оператор IN.

1. Текст скрипта:

select "BusinessEntityID",

"Name",

"SalesPersonID"

from "Sales"."Store" s

where "Name" in ('Next-Door Bike Store',

'Area Bike Accessories',

'Top of the Line Bikes',

'Valley Toy Store',

'Global Plaza');

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

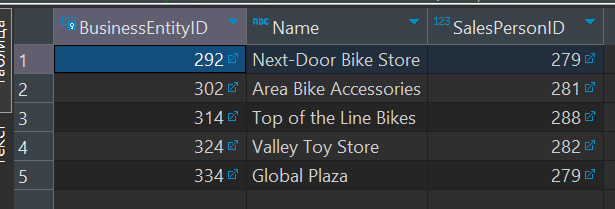


Рисунок 6 − Результат выполнения шестого задания

Задание 7 и 10

1. 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.

1. Текст скрипта:

select "BusinessEntityID",

"AccountNumber",

"Name",

coalesce ("PurchasingWebServiceURL",'not available')"PurchasingWebServiceURL"

from "Purchasing"."Vendor" v ;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

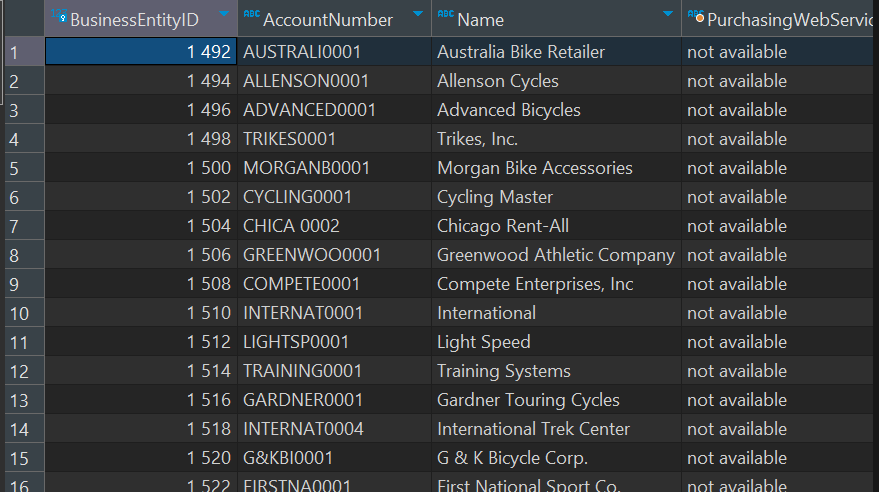


Рисунок 7 − Результат выполнения седьмого и десятого заданий

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 Style 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 ("Color","Style",'UNKNOWN') as Meauserment

from "Production"."Product" p

;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

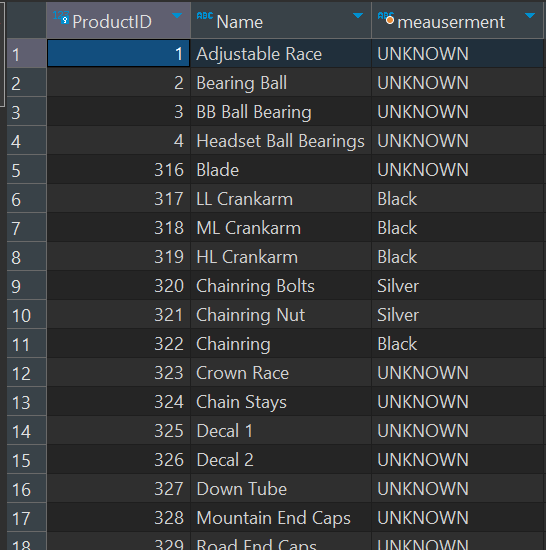


Рисунок 8 − Результат выполнения восьмого задания

Задание 9

1. Формулировка задания:

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

1. Текст скрипта:

select "Name"

from "Production"."Culture" c

order by "Name";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).

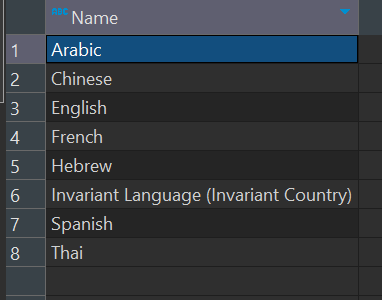


Рисунок 9 − Результат выполнения восьмого задания

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

Подтверждение о правильно выполненной работе представлено ниже (Рисунок 9).



Рисунок 10 − Подтверждение о правильно выполненной работе