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 1

Выполнил: ст. гр. ТКИ-442

Белов Сергей Владимирович

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.Document.

1. Текст скрипта:

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

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

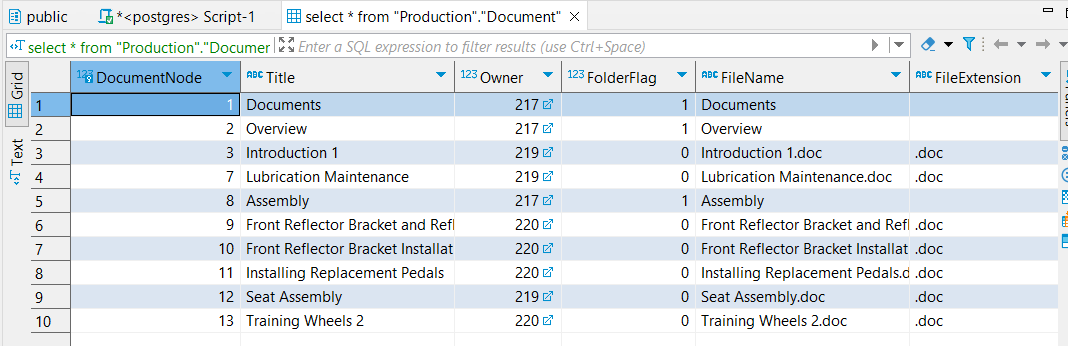


Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

Показать поля ProductID, DocumentNode из таблицы Production.ProductDocument.

1. Текст скрипта:

SELECT "ProductID", "DocumentNode" FROM "Production"."ProductDocument";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

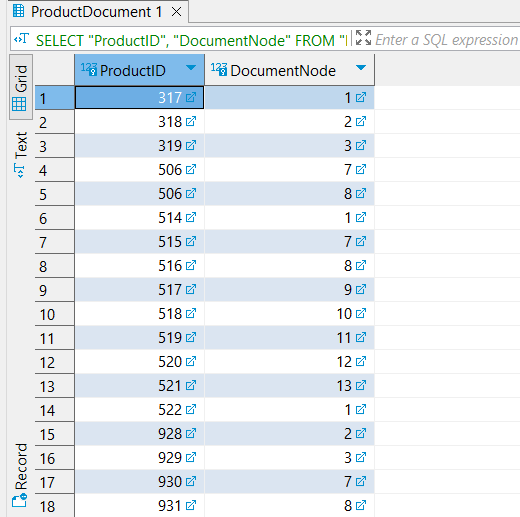


Рисунок 2 − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Показать поля ProductID, BusinessEntityID, AverageLeadTime, StandardPrice из таблицы Purchasing.ProductVendor, где средний промежуток времени (в днях) между размещением заказа у продавца и получением приобретенного продукта (AverageLeadTime) больше 16, а обычная отпускная цена продавца (StandardPrice) находится в диапазоне от $50 до $60. Использовать оператор BETWEEN.

1. Текст скрипта:

SELECT "ProductID", "BusinessEntityID", "AverageLeadTime", "StandardPrice" FROM "Purchasing"."ProductVendor" WHERE "AverageLeadTime" > 16

AND "StandardPrice" BETWEEN 50 AND 60;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

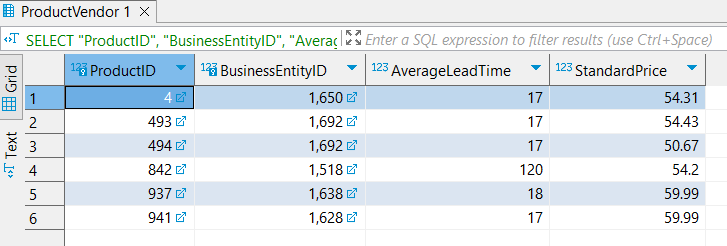


Рисунок 3 − Результат выполнения первого задания

Задание 4

1. Формулировка задания:

Показать поля PurchaseOrderID, RevisionNumber, Status, ShipDate из таблицы Purchasing.PurchaseOrderHeader, для которых известна ориентировочная дата отгрузки от поставщика (ShipDate) и текущее состояние заказа (Status) "Ожидание" (= 1).

1. Текст скрипта:

SELECT "PurchaseOrderID", "RevisionNumber", "Status", "ShipDate" FROM "Purchasing"."PurchaseOrderHeader"

WHERE "ShipDate" IS NOT NULL AND "Status" = 1;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

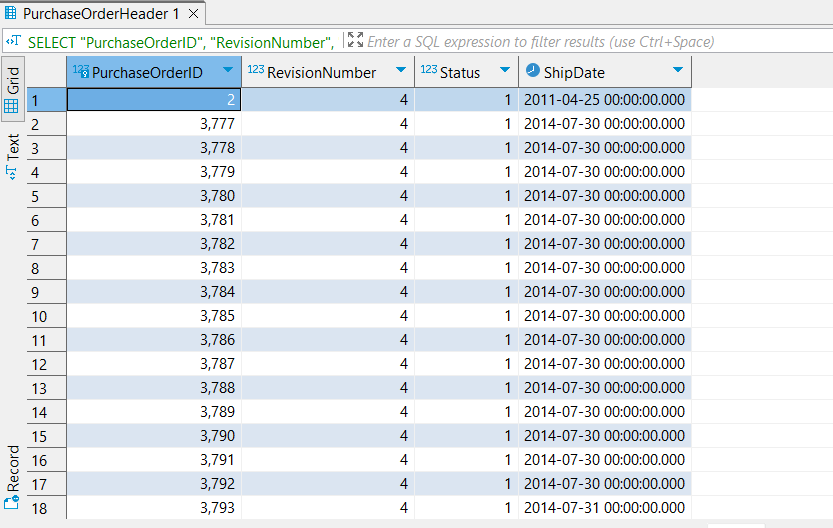


Рисунок 4 − Результат выполнения первого задания

Задание 5

1. Формулировка задания:

Показать cтандартный код ISO для стран и регионов (CountryRegionCode) и название страны или региона (Name), имеющие отношение к Конго (Congo) из таблицы Person.CountryRegion.

1. Текст скрипта:

SELECT "CountryRegionCode", "Name" FROM "Person"."CountryRegion" WHERE "Name" LIKE '%Congo%';

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

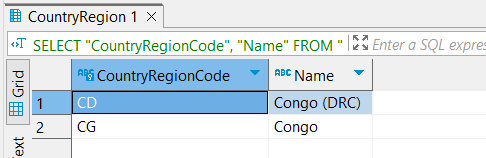


Рисунок 5 − Результат выполнения первого задания

Задание 6

1. Формулировка задания:

Показать поля AddressID, AddressLine1, City из таблицы Person.Address. Показать только адреса из списка городов (Bothell, Dallas, Ottawa, Calgary). Использовать оператор IN.

1. Текст скрипта:

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

FROM "Person"."Address"

WHERE "City" IN ('Bothell', 'Dallas', 'Ottawa', 'Calgary');

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

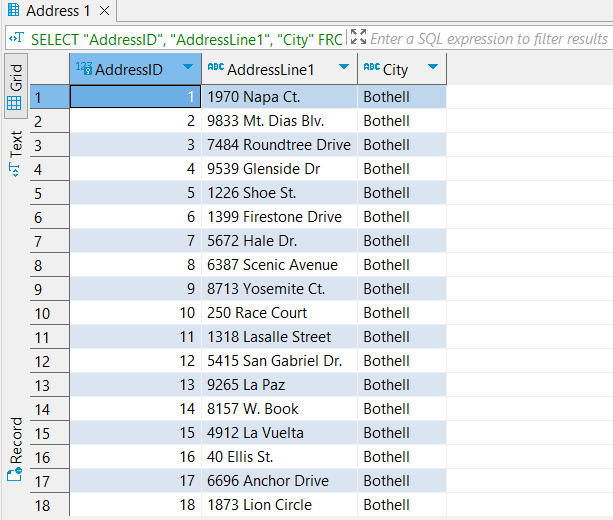


Рисунок 6 − Результат выполнения первого задания

Задание 7

1. Формулировка задания:

Показать поля BusinessEntityID, EmailAddressID и EmailAddress из таблицы Person.EmailAddress. Неизвестные значения поля EmailAddress заменить на 'no email'.

1. Текст скрипта:

SELECT "BusinessEntityID", "EmailAddressID", COALESCE("EmailAddress", 'no email') AS EmailAddress

FROM "Person"."EmailAddress";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

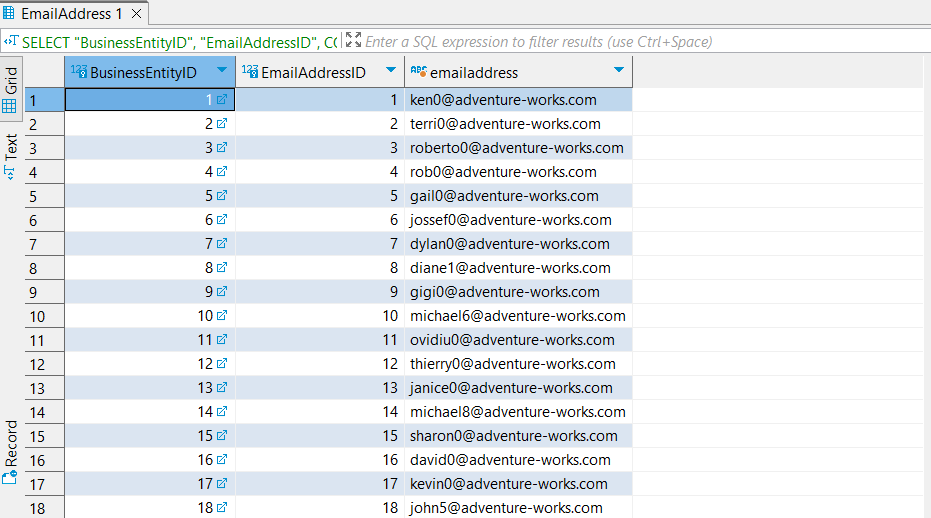


Рисунок 7 − Результат выполнения первого задания

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 известно Class, то показать его, а иначе, показать значение в поле Style. Если и в поле Style значение неизвестно, то вывести значение 'UNKNOWN'.

1. Текст скрипта:

SELECT "ProductID", "Name", COALESCE("Class", "Style", 'UNKNOWN') AS Measurement

FROM "Production"."Product";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

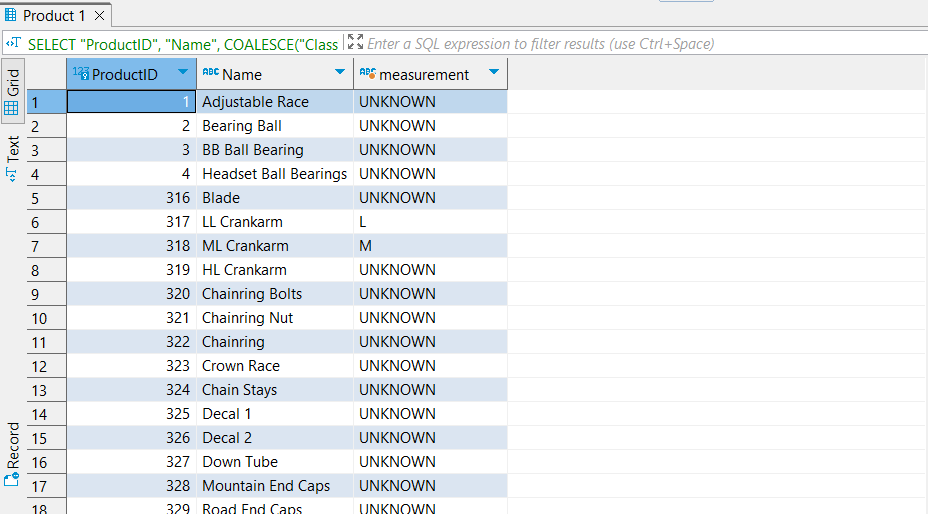


Рисунок 8 − Результат выполнения первого задания

Задание 9

1. Формулировка задания:

Отсортировать строки таблицы Person.AddressType в алфавитном порядке наименования описания типа адреса (Name). Показать поля AddressTypeID и Name.

1. Текст скрипта:

SELECT "AddressTypeID", "Name"

FROM "Person"."AddressType"

ORDER BY "Name";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).

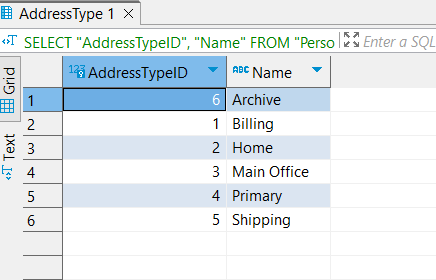


Рисунок 9 − Результат выполнения первого задания

Задание 10

1. Формулировка задания:

Заменить на NULL значение в поле цвет (Color), если цвет товара красный (red). Показать поля ProductID, Name, Color из таблицы Production.Product, если цвет товара определен (поле не пустое).

1. Текст скрипта:

SELECT "ProductID", "Name", NULLIF("Color", 'red') AS Color

FROM "Production"."Product"

WHERE "Color" IS NOT NULL;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10 ).

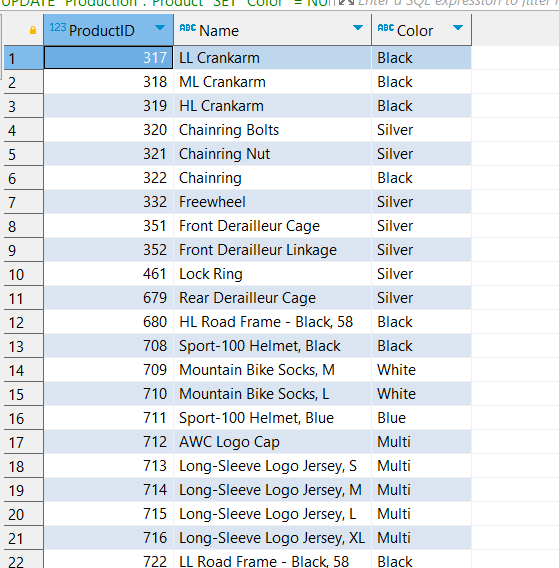


Рисунок 10 − Результат выполнения первого задания

## 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