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 №1

по дисциплине:

«Информационное обеспечение систем управления»

на тему:

«Фильтрация набор данных»

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 фильтрации набора данных. Научиться создавать простые запросы на фильтрацию данных.

**Результаты выполненной работы**

**Задание** **1**

Формулировка задания:

Показать все поля из таблицы Production.ScrapReason.

Текст скрипта:

**select** \*

**from** "Production"."ScrapReason" *sr*

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

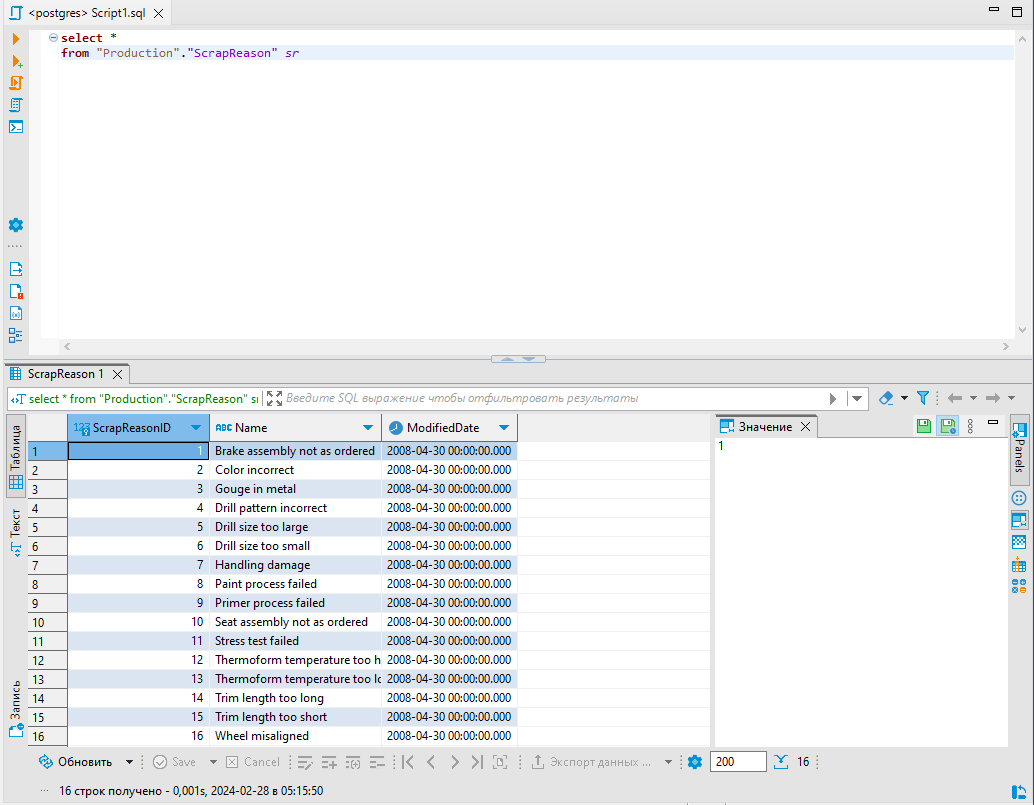


Рисунок 1 – Результат выполнения первого задания

**Задание** **2**

Формулировка задания:

Показать поля (ProductPhotoID, ThumbnailPhotoFileName, LargePhotoFileName) из таблицы Production.ProductPhoto.

Текст скрипта:

**select** *pp*."ProductPhotoID"

,*pp*."ThumbnailPhotoFileName"

,*pp*."LargePhotoFileName"

**from** "Production"."ProductPhoto" *pp*

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

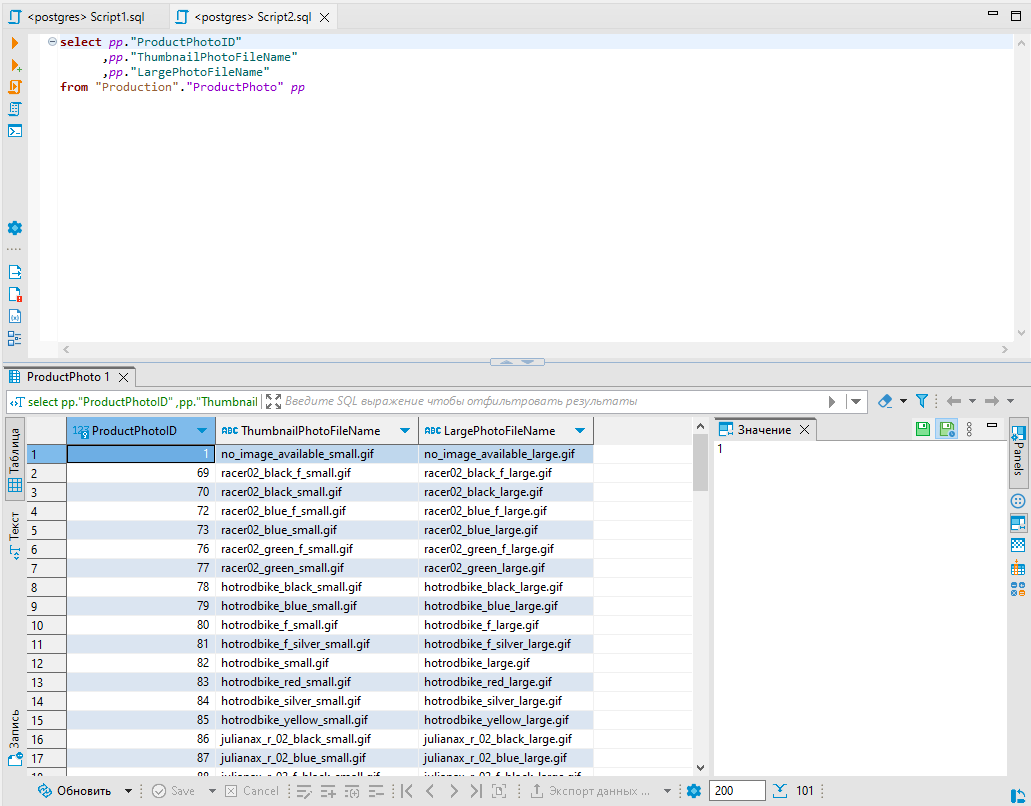


Рисунок 2 – Результат выполнения второго задания

**Задание** **3**

Формулировка задания:

Показать поля TerritoryID, Name, Group, SalesYTD из таблицы Sales.SalesTerritory. Показать только те записи, для которых продажи на территории с начала года (SalesYTD) больше $3000000 и меньше $5000000. Использовать оператор BETWEEN.

Текст скрипта:

**select** *st*."TerritoryID"

,*st*."Name"

,*st*."Group"

,*st*."SalesYTD"

**from** "Sales"."SalesTerritory" *st*

**where** *st*."SalesYTD" **between** 3000000 **and** 5000000;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

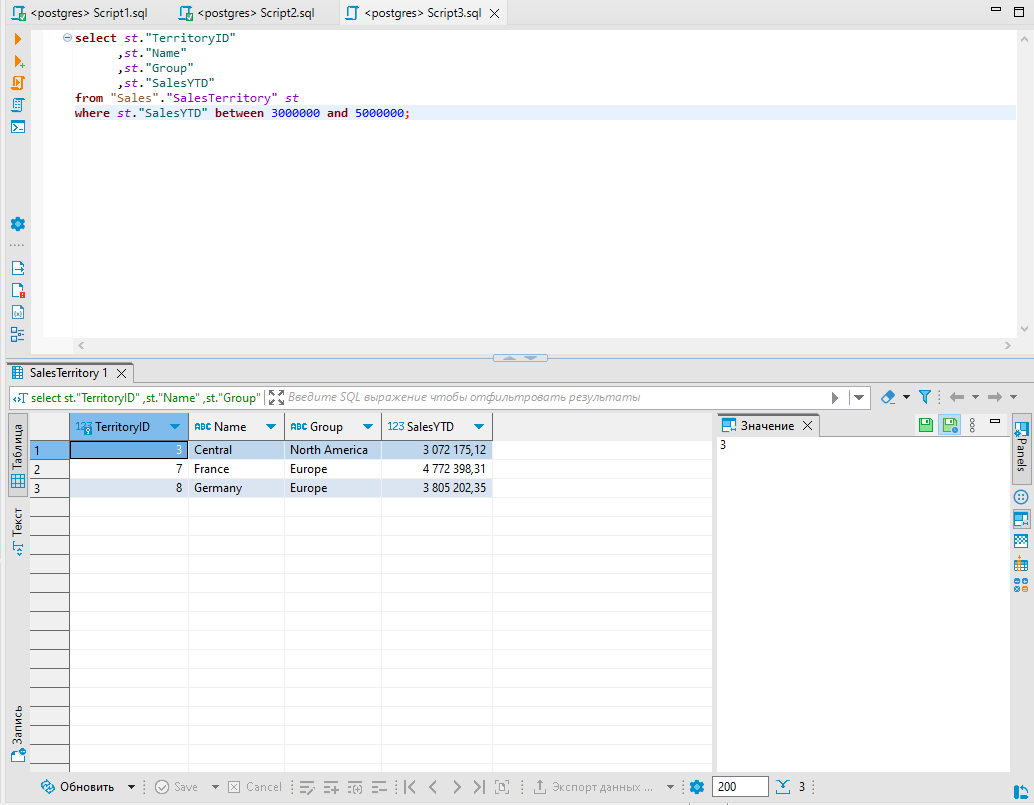


Рисунок 3 – Результат выполнения третьего задания

**Задание** **4**

Формулировка задания:

Из таблицы Sales.Store показать ID сотрудника (BusinessEntityID), название магазина (Name) и ID продавца (SalesPersonID). Показать только существующие ID продавцов.

Текст скрипта:

**select** *s*."BusinessEntityID"

,*s*."Name"

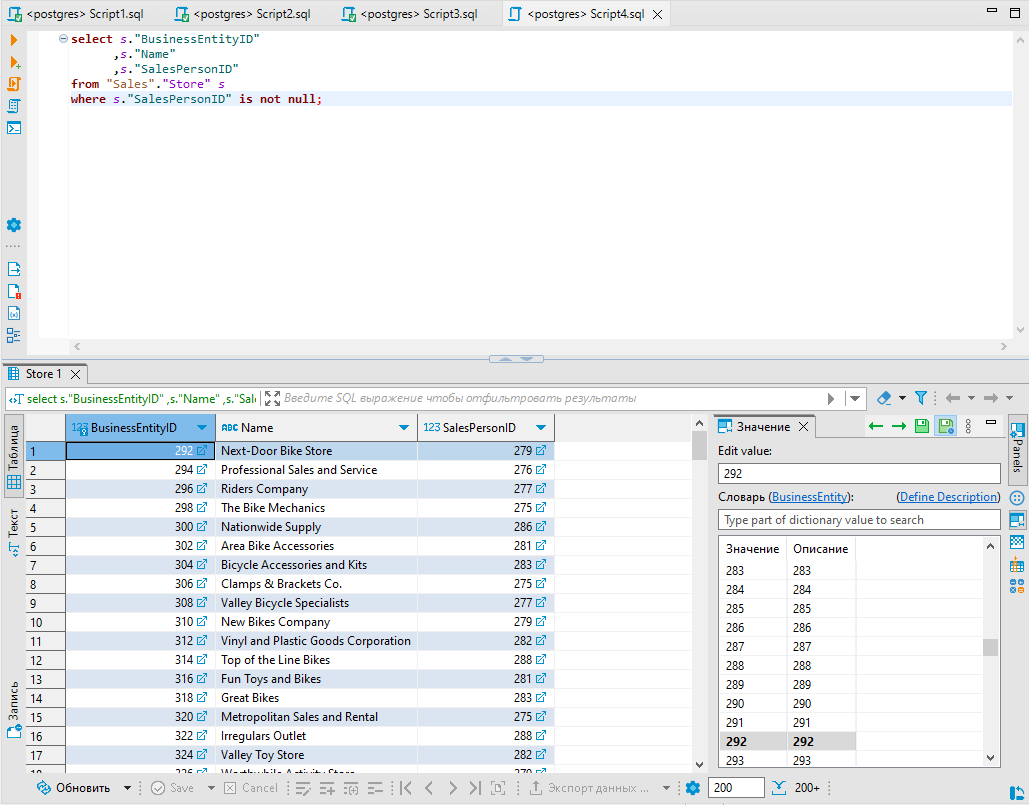
,*s*."SalesPersonID"

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

**where** *s*."SalesPersonID" **is** **not** **null**;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

Рисунок 4 − Результат выполнения четвертого задания

**Задание** **5**

Формулировка задания:

Показать место документа в памяти (DocumentNode), заголовок документа (Title) и расширение файла (FileExtension) из таблицы Production.Document. Показать только файлы с расширением '.doc'.

Текст скрипта:

**select** *d*."DocumentNode"

,*d*."Title"

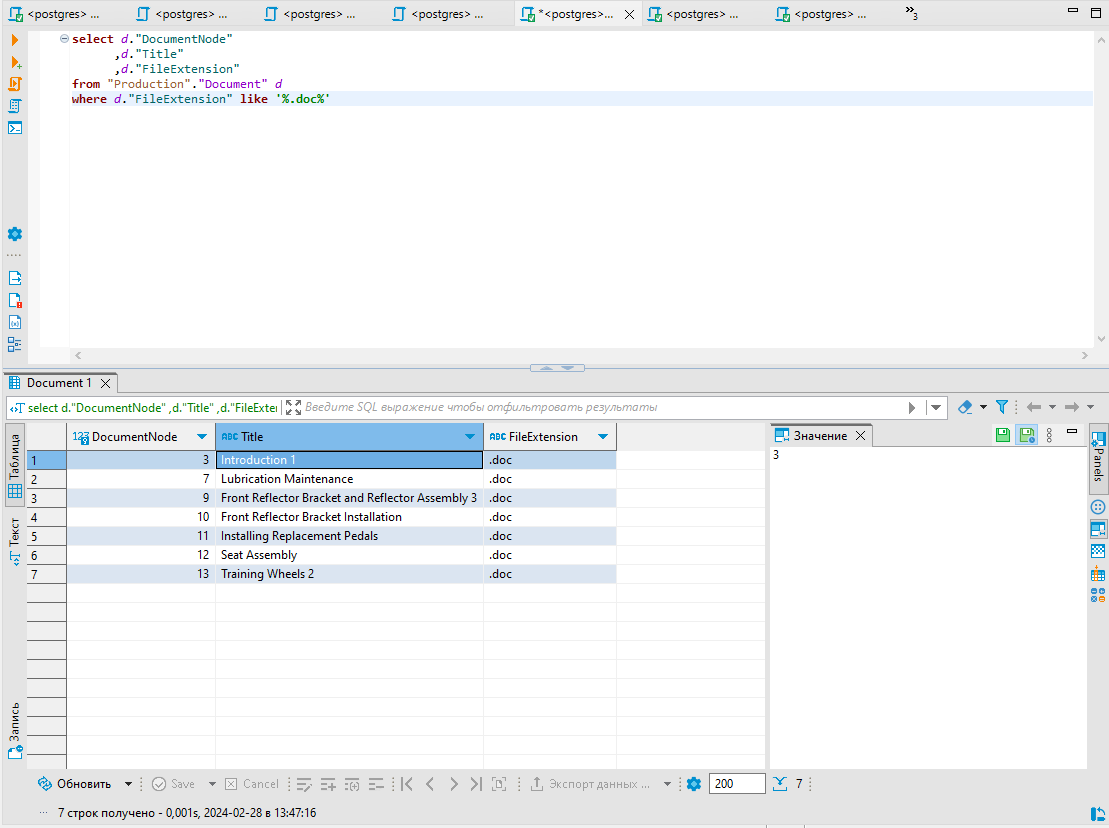
,*d*."FileExtension"

**from** "Production"."Document" *d*

**where** *d*."FileExtension" **like** **'%.doc%'**

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

Рисунок 5 − Результат выполнения пятого задания

**Задание** **6**

Формулировка задания:

Из таблицы Production.ProductSubcategory показать поля ProductSubcategoryID, Name. Показать только значения поля Name из списка «переключатели», «гарнитуры», «колеса» ('Derailleurs', 'Headsets', 'Wheels'). Использовать оператор IN.

Текст скрипта:

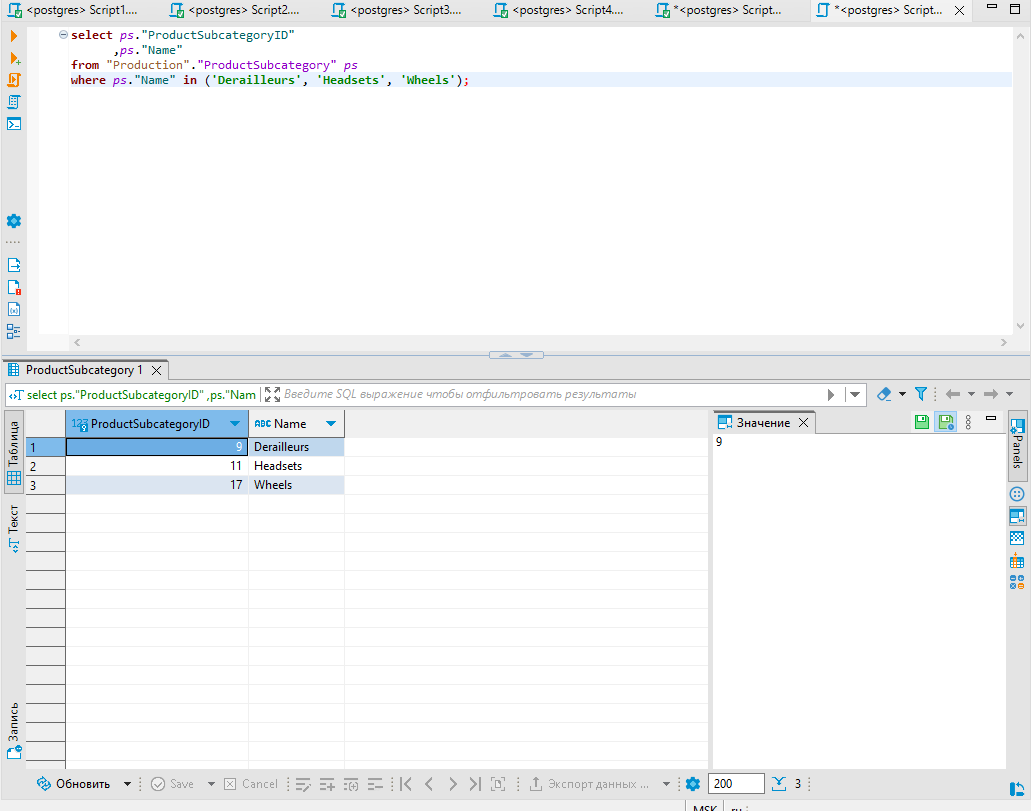
**select** *ps*."ProductSubcategoryID"

,*ps*."Name"

**from** "Production"."ProductSubcategory" *ps*

**where** *ps*."Name" **in** (**'Derailleurs'**, **'Headsets'**, **'Wheels'**);

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6). Рисунок 6 − Результат выполнения шестого задания

**Задание** **7**

Формулировка задания:

Показать поля BillOfMaterialsID, ProductAssemblyID, ComponentID, UnitMeasureCode из таблицы Production.BillOfMaterials. Если значение поля ProductAssemblyID – NULL, заменить его на 0 и выделить в отдельное поле с названием ProductAssemblyID\_0.

Текст скрипта:

**select** *bom2*."BillOfMaterialsID"

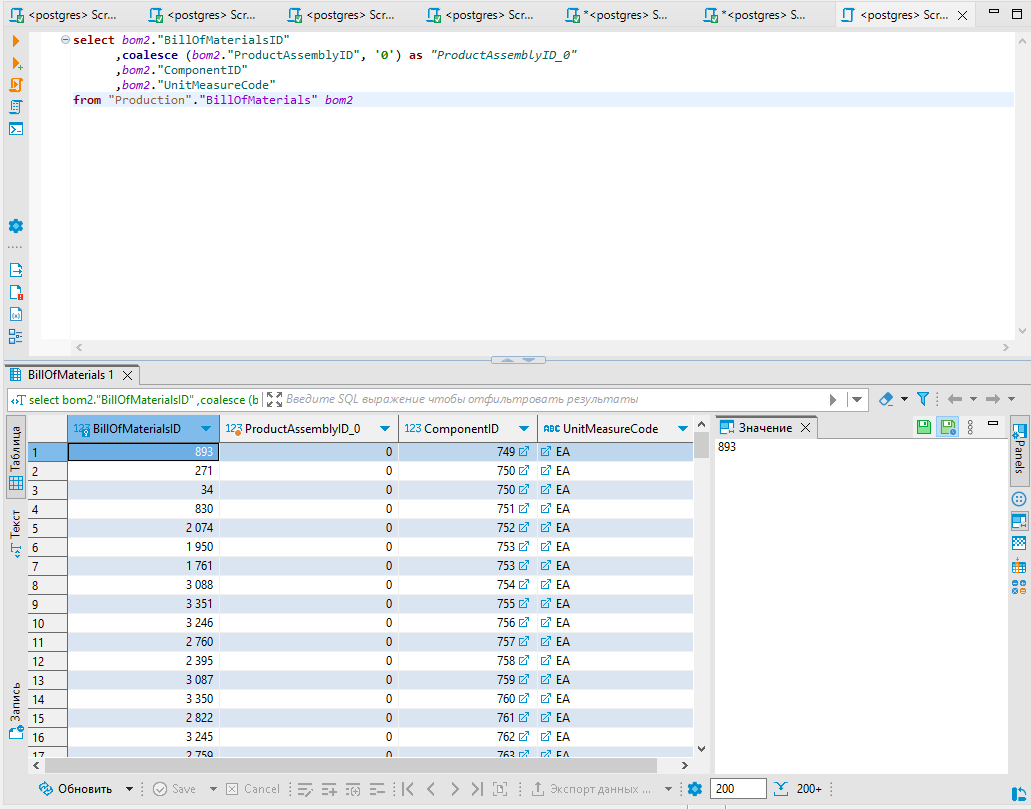
,**coalesce** (*bom2*."ProductAssemblyID", **'0'**) **as** *"ProductAssemblyID\_0"*

,*bom2*."ComponentID"

,*bom2*."UnitMeasureCode"

**from** "Production"."BillOfMaterials" *bom2*

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 Measurement, так, чтобы, если значение в поле Color известно, то показать его, а иначе, показать значение в поле Style. Если и в поле Style значение неизвестно, то вывести значение 'UNKNOWN'.

Текст скрипта:

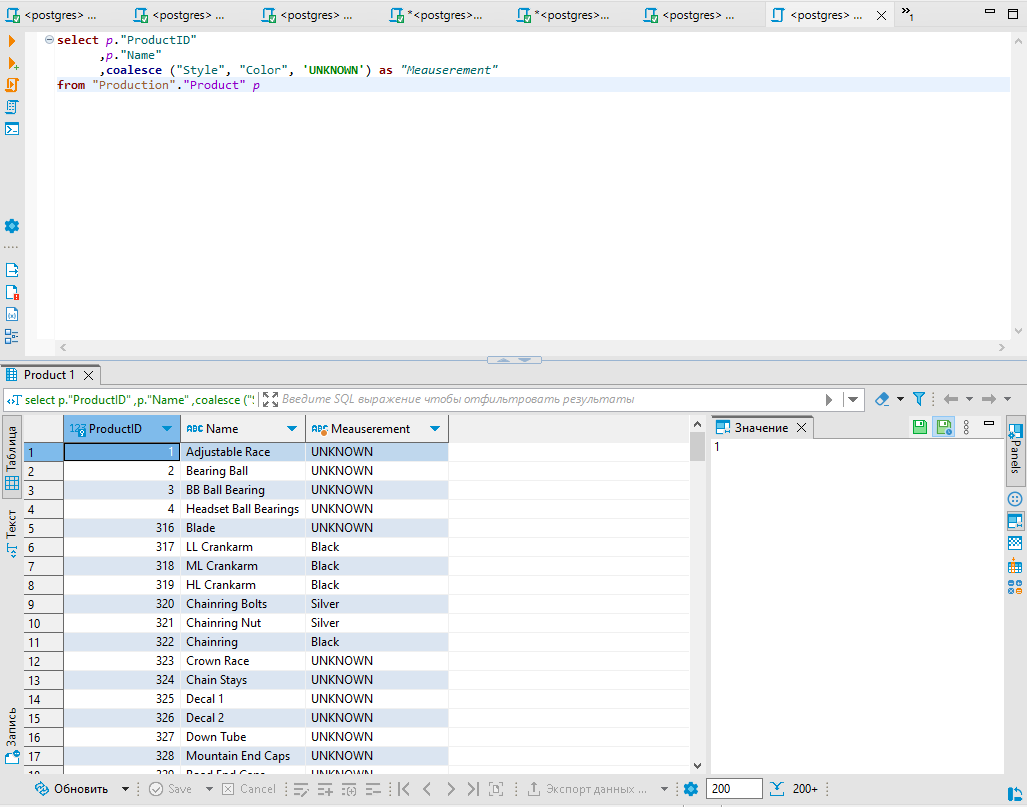
**select** *p*."ProductID"

,*p*."Name"

,**coalesce** ("Style", "Color", **'UNKNOWN'**) **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.Document показать поля DocumentNode, Title и FileExtension. Отсортировать строки в алфавитном порядке в соответствии с полем Title.

Текст скрипта:

**select** *d*."DocumentNode"

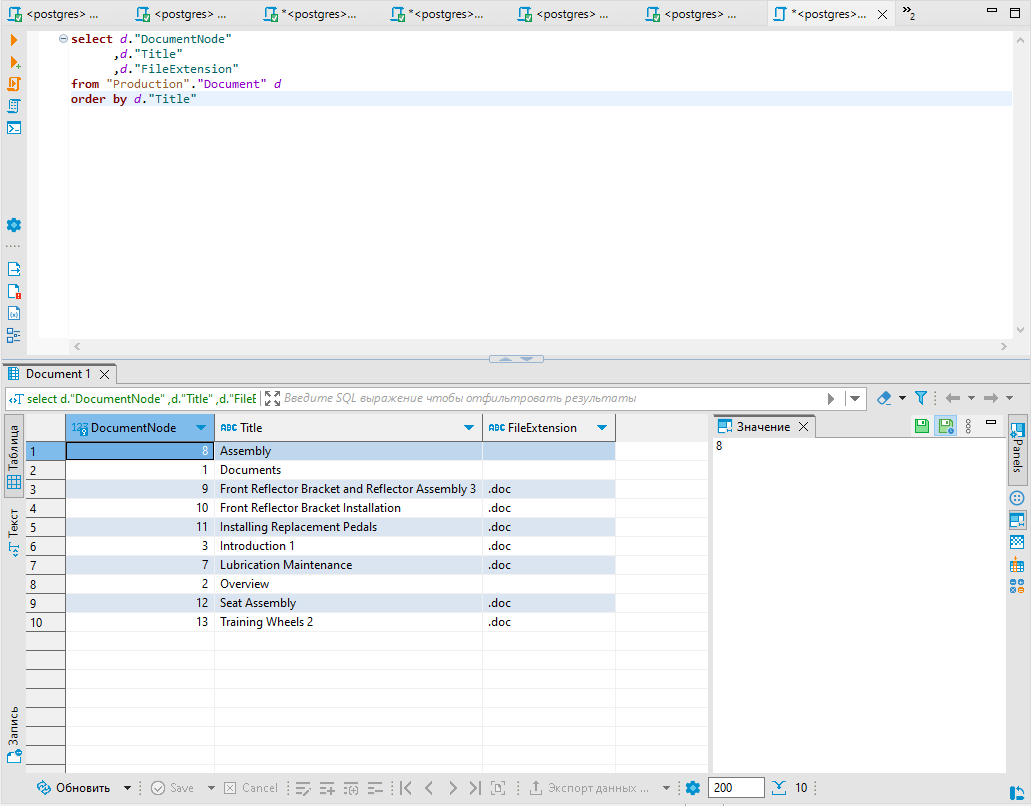
,*d*."Title"

,*d*."FileExtension"

**from** "Production"."Document" *d*

**order** **by** *d*."Title"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9). Рисунок 9 − Результат выполнения девятого задания

**Задание** **10**

Формулировка задания:

Показать поля BusinessEntityID, NationalIDNumber, LoginID, OrganizationNode из таблицы HumanResources.Employee. Показать только существующие значения поля OrganizationNode. Если поле OrganizationNode имеет значение 0x5AE358, то нужно обратить это значение в NULL. Названия полей оставить без изменений.

Текст скрипта:

**select** *e*."BusinessEntityID"

,*e*."NationalIDNumber"

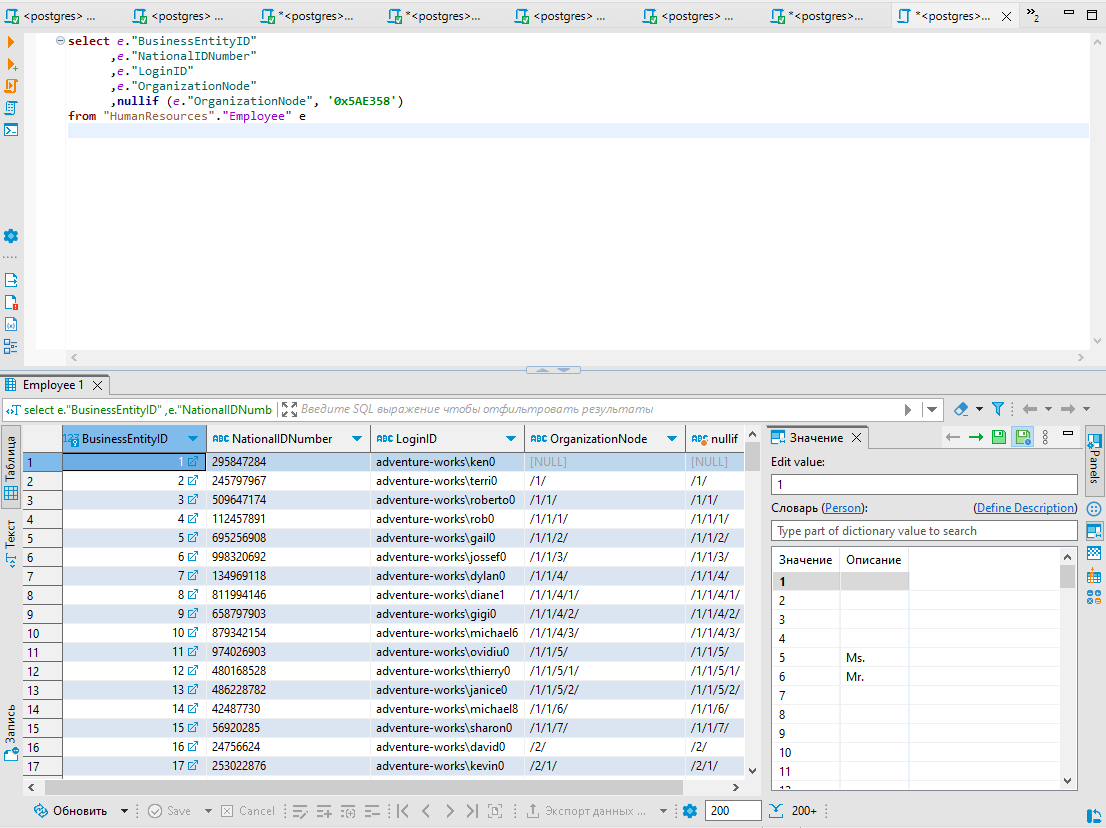
,*e*."LoginID"

,*e*."OrganizationNode"

,**nullif** (*e*."OrganizationNode", **'0x5AE358'**)

**from** "HumanResources"."Employee" e

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10). Рисунок 10 − Результат выполнения десятого задания

Вывод: изучены операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для фильтрации набора данных, а также научились создавать простые запросы на фильтрацию данных.