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 19

Выполнил: ст. гр. ТКИ-441

Чекан Фёдор Сергеевич

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.ProductReview.

1. Текст скрипта:

-- Показать все поля из таблицы Production.ProductReview.

select \*

from "Production"."ProductReview";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

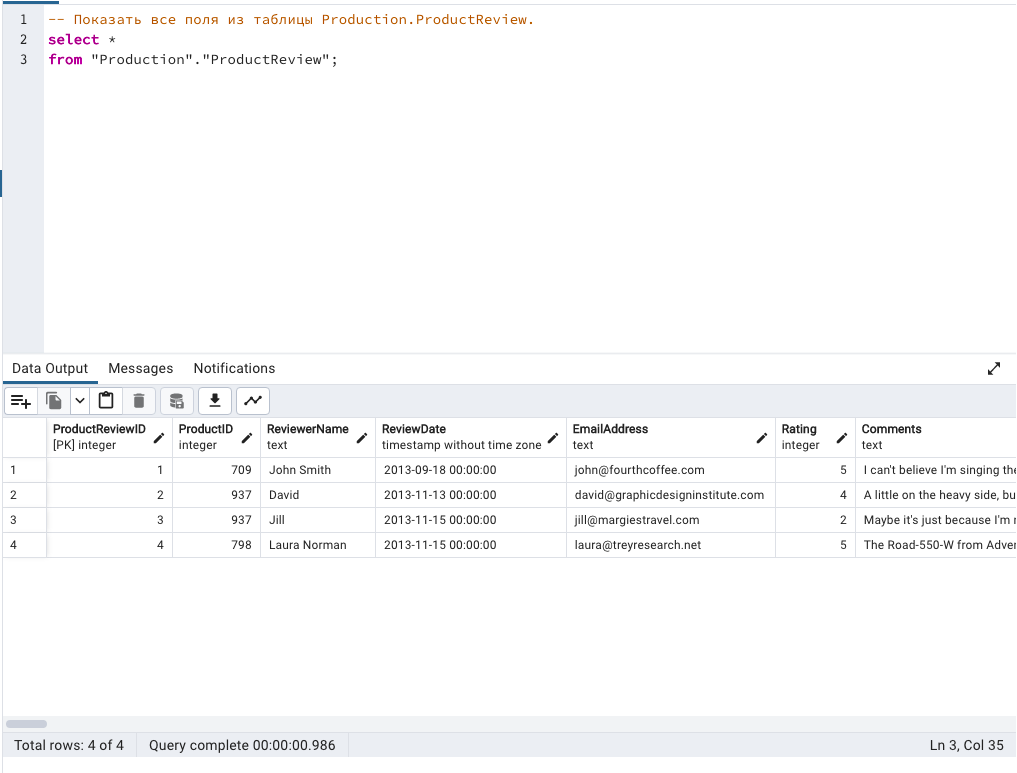


Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

Показать поля ScrapReasonID, Name из таблицы Production.ScrapReason.

1. Текст скрипта:

--Показать поля ScrapReasonID, Name из таблицы

--Production.ScrapReason.

select "ScrapReasonID", "Name"

from "Production"."ScrapReason";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

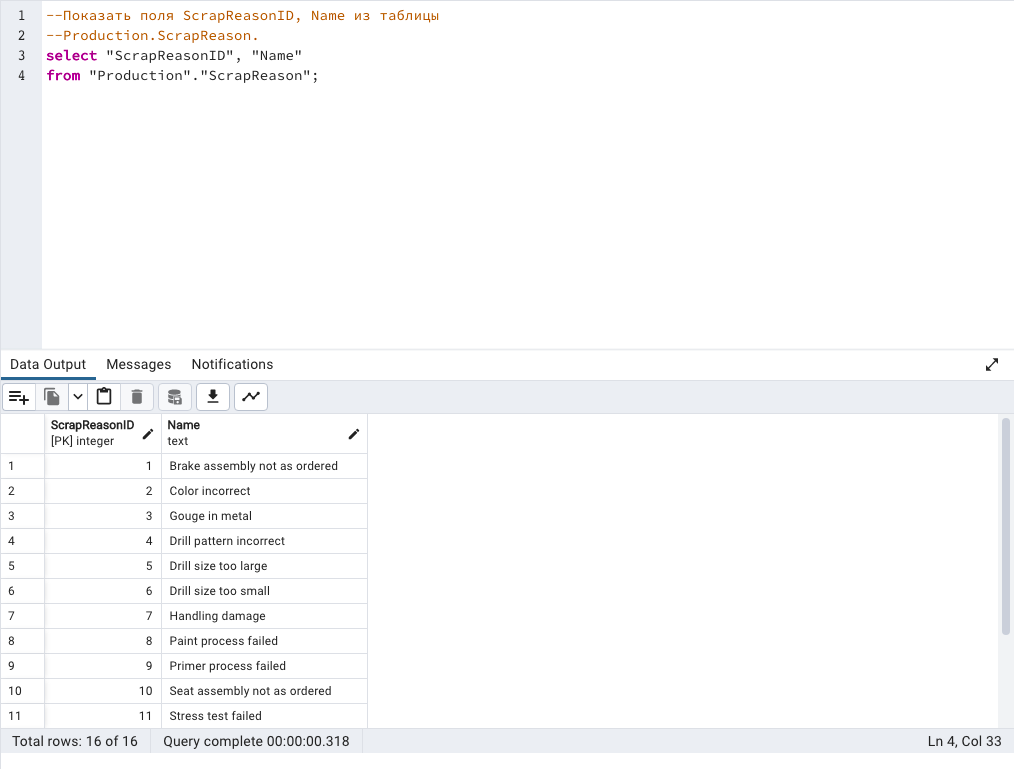


Рисунок 2 − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Из таблицы Sales.CurrencyRate показать ID курса валюты (CurrencyRateID), даты, в которые проверяется курс валюты (CurrencyRateDate) и конечный курс валюты на данную дату (EndOfDayRate), взятый от $1 до $10. Использовать оператор BETWEEN.

1. Текст скрипта:

--Из таблицы Sales.CurrencyRate показать ID курса валюты

--(CurrencyRateID), даты, в которые проверяется курс валюты

--(CurrencyRateDate) и конечный курс валюты на данную дату

--(EndOfDayRate), взятый от $1 до $10. Использовать оператор

--BETWEEN.

select "CurrencyRateID", "CurrencyRateDate", "EndOfDayRate"

from "Sales"."CurrencyRate"

where "EndOfDayRate" between 1 and 10;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

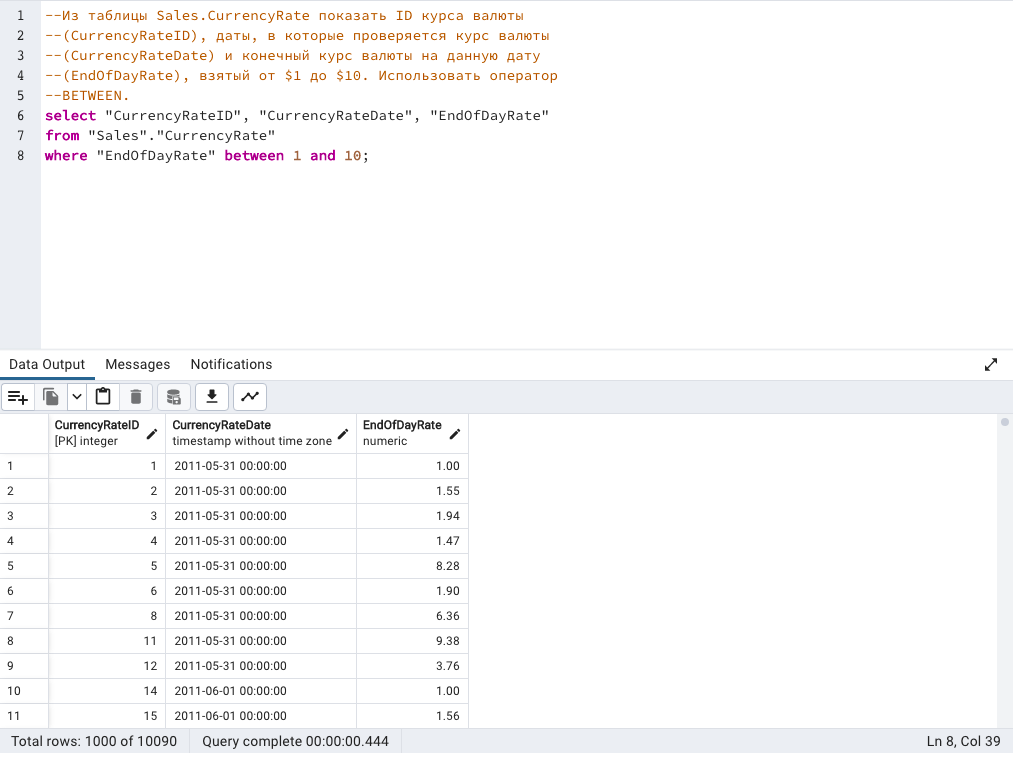


Рисунок 3 − Результат выполнения третьего задания

Задание 4

1. Формулировка задания:

Показать цену на товар (ListPrice), дату начала действия цены (StartDate) и дату окончания действия цены (EndDate) из таблицы Production.ProductListPriceHistory. Показать только те цены, которые актуальны (EndDate = NULL).

1. Текст скрипта:

--Показать цену на товар (ListPrice), дату начала действия цены

--(StartDate) и дату окончания действия цены (EndDate) из таблицы

--Production.ProductListPriceHistory. Показать только те

--цены, которые актуальны (EndDate = NULL).

select "ListPrice", "StartDate", "EndDate"

from "Production"."ProductListPriceHistory"

where "EndDate" is null;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

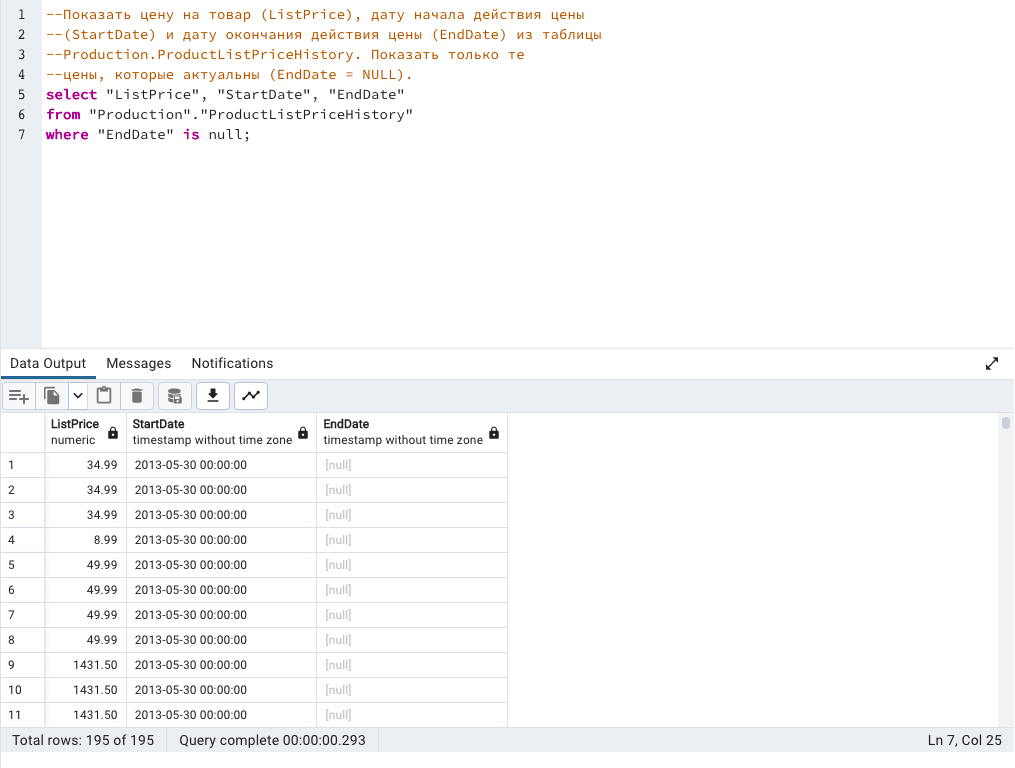


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

1. Формулировка задания:

Показать все описания (Name) неисправностей (damage). Таблица Production.ScrapReason.

1. Текст скрипта:

--Показать все описания (Name) неисправностей (damage). Таблица

--Production.ScrapReason.

select \*

from "Production"."ScrapReason"

where "Name" like '%damage%';

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

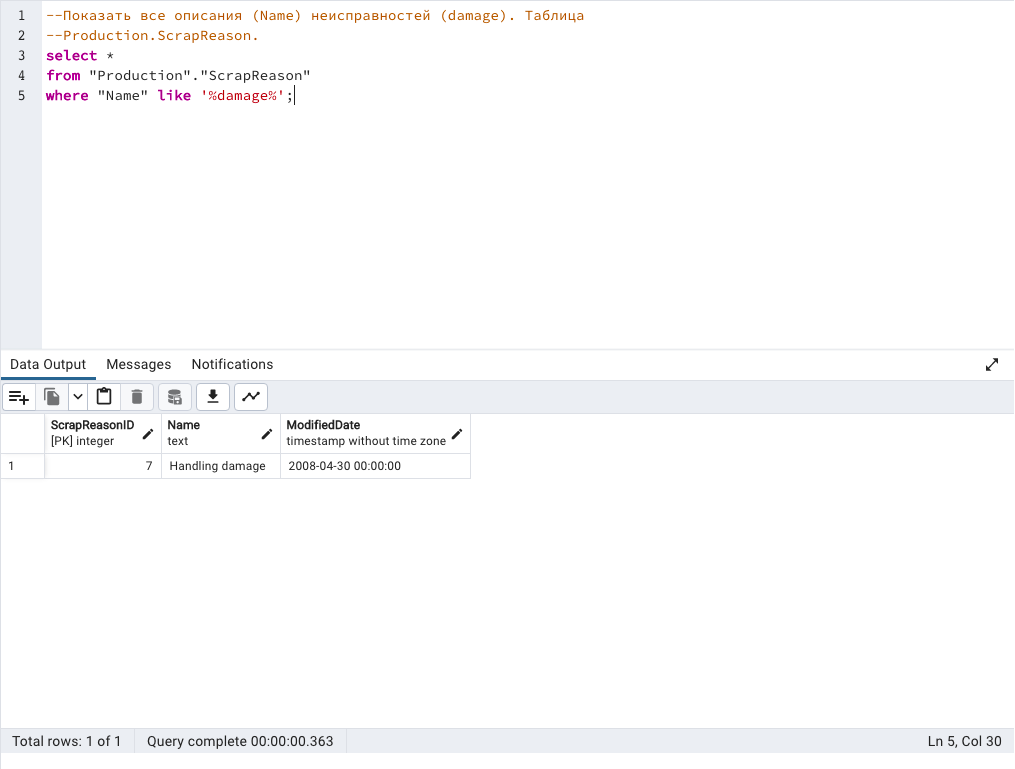


Рисунок 5 − Результат выполнения пятого задания

Задание 6

1. Формулировка задания:

Из таблицы Production.ProductInventory показать поля ProductID, Shelf и Quantity. Вывести только те значения поля Shelf, которые принадлежат списку ('E', 'H', 'R', 'T'). Использовать оператор IN.

1. Текст скрипта:

--Из таблицы Production.ProductInventory показать поля

--ProductID, Shelf и Quantity. Вывести только те значения поля

--Shelf, которые принадлежат списку ('E', 'H', 'R', 'T').

--Использовать оператор IN.

select "ProductID", "Shelf", "Quantity"

from "Production"."ProductInventory"

where "Shelf" in ('E', 'H', 'R', 'T');

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

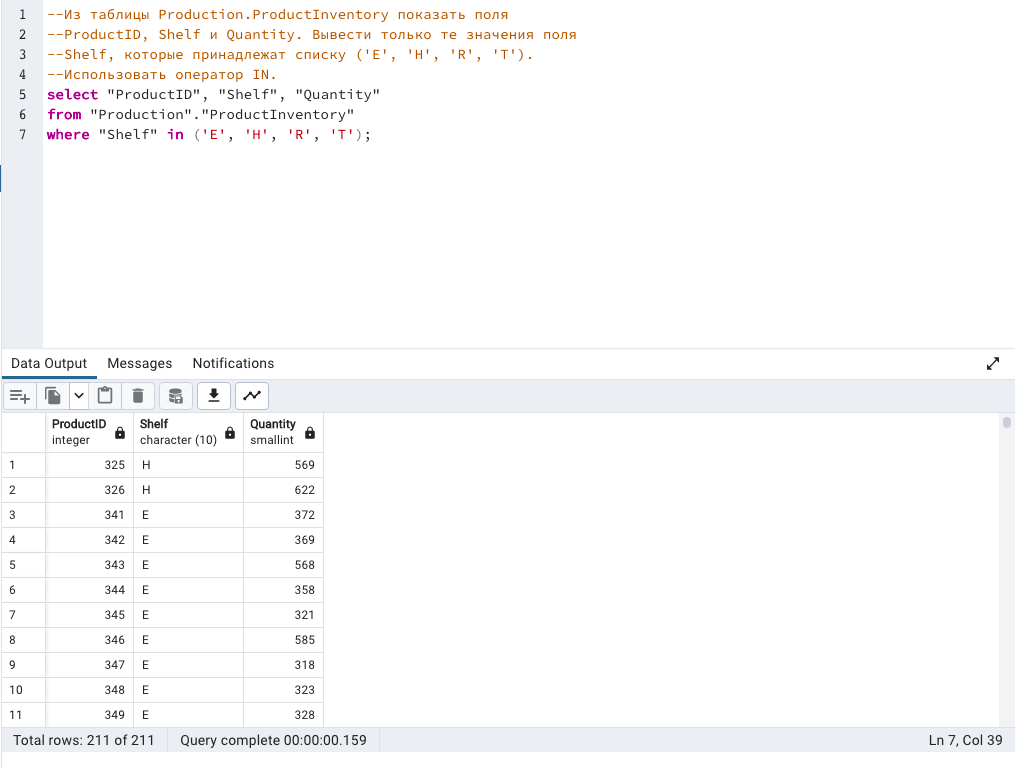


Рисунок 6 − Результат выполнения шестого задания

Задание 7

1. Формулировка задания:

Показать ID организации (BusinessEntityID) и ID её терииториального расположения (TerritoryID) из таблицы Sales.SalesPerson. Все неизвестные значения поля TerritoryID заменить на 0. Названия полей оставить без изменений.

1. Текст скрипта:

--Показать ID организации (BusinessEntityID) и ID её

--терииториального расположения (TerritoryID) из таблицы

--Sales.SalesPerson. Все неизвестные значения поля TerritoryID

--заменить на 0. Названия полей оставить без изменений.

select "BusinessEntityID", coalesce("TerritoryID", 0) as "TerritoryID"

from "Sales"."SalesPerson";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

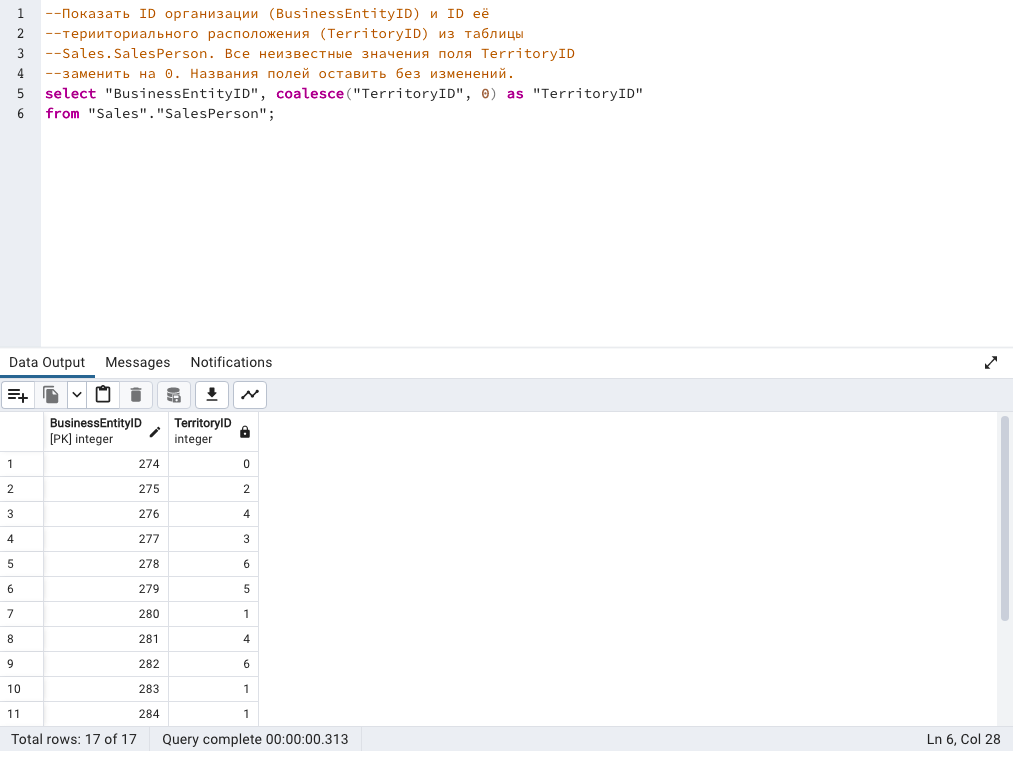


Рисунок 7 − Результат выполнения седьмого задания

Задание 8

1. 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 Weight. Если и в поле Weight значение неизвестно, то вывести значение 'UNKNOWN'.

1. Текст скрипта:

--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 Weight. Если и в поле Weight

--значение неизвестно, то вывести значение 'UNKNOWN'.

select "ProductID", "Name", coalesce("Color", cast("Weight" as text), 'UNKNOWN') as "Measurement"

from "Production"."Product";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

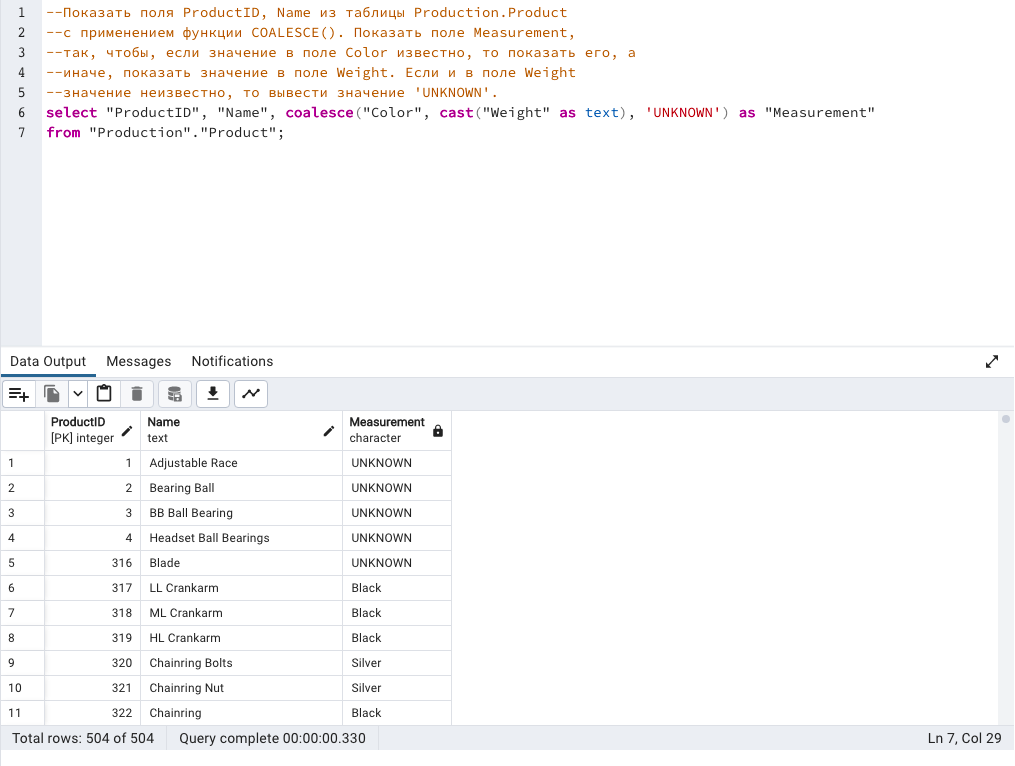


Рисунок 8 − Результат выполнения восьмого задания

Задание 9

1. Формулировка задания:

Отсортировать все поля (BusinessEntityID, RateChangeDate, Rate) таблицы HumanResources.EmployeePayHistory в порядке возрастания относительно поля Rate.

1. Текст скрипта:

--Отсортировать все поля (BusinessEntityID, RateChangeDate,

--Rate) таблицы HumanResources.EmployeePayHistory в порядке

--возрастания относительно поля Rate.

select "BusinessEntityID", "RateChangeDate", "Rate"

from "HumanResources"."EmployeePayHistory"

order by "Rate" ASC;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).

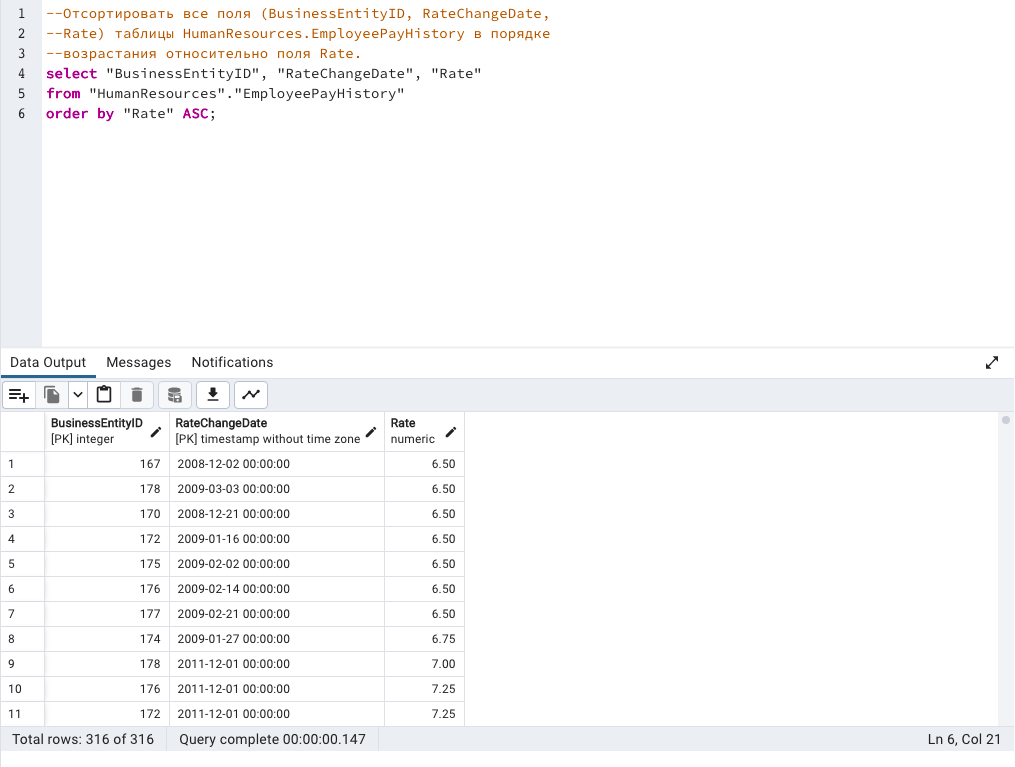


Рисунок 9 − Результат выполнения девятого задания

Задание 10

1. Формулировка задания:

Из таблицы Person.BusinessEntityContact показать поля PersonID и ContactTypeID. Все значения поля ContactTypeID, равные 11, заменить на NULL и вынести в отдельное поле с названием NullContactTypeID.

1. Текст скрипта:

--Из таблицы Person.BusinessEntityContact показать поля

--PersonID и ContactTypeID. Все значения поля ContactTypeID,

--равные 11, заменить на NULL и вынести в отдельное поле с названием

--NullContactTypeID.

select "PersonID", "ContactTypeID", nullif("ContactTypeID", 11) as "NullContactTypeID"

from "Person"."BusinessEntityContact";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10).

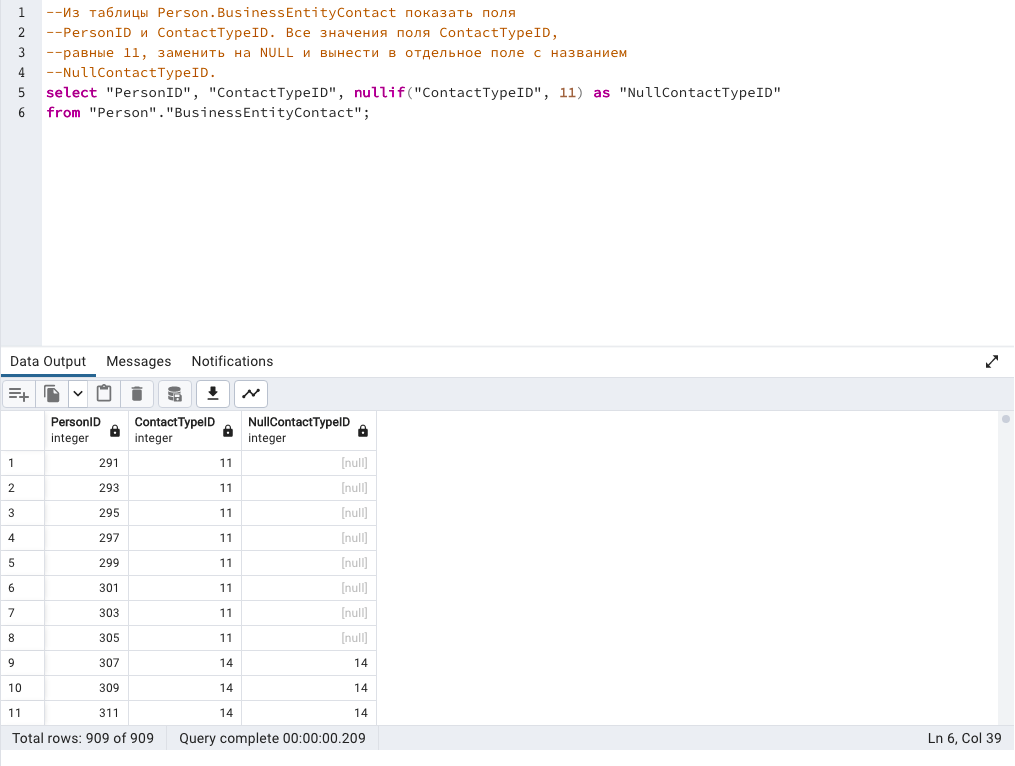


Рисунок 10 − Результат выполнения десятого задания

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

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

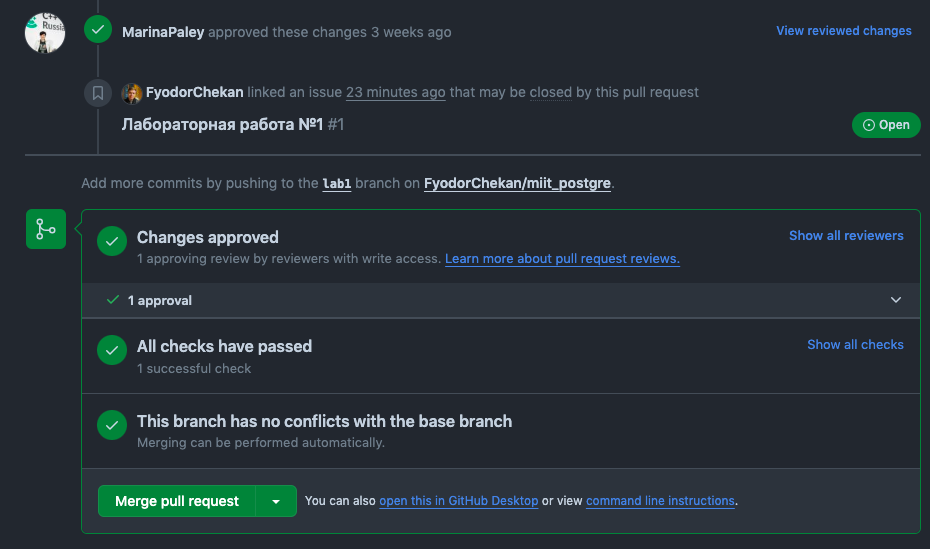


Рисунок 11 − Подтверждение о правильно выполненной работе