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 №3

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 диалекта PostgreSQL, необходимые для работы с данными, находящимися в разных таблицах. Научиться создавать запросы на соединение данных из множества таблиц. Отчет по выполненной работе должен быть выполнен с соблюдением ГОСТ по НИР 7.32.

## РЕЗУЛЬТАТЫ ВЫПОЛНЕННОЙ РАБОТЫ

Задание 1

1. Формулировка задания:

Показать названия единиц измерения веса товаров и названия товаров (поля Name) из таблиц Production.Product, Production.UnitMeasure.

1. Текст скрипта:

--Показать названия единиц измерения веса товаров и названия товаров

--(поля Name) из таблиц Production.Product,

--Production.UnitMeasure.

select p."Name", u."Name"

from "Production"."Product" p

inner join "Production"."UnitMeasure" u on p."WeightUnitMeasureCode" = u."UnitMeasureCode";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

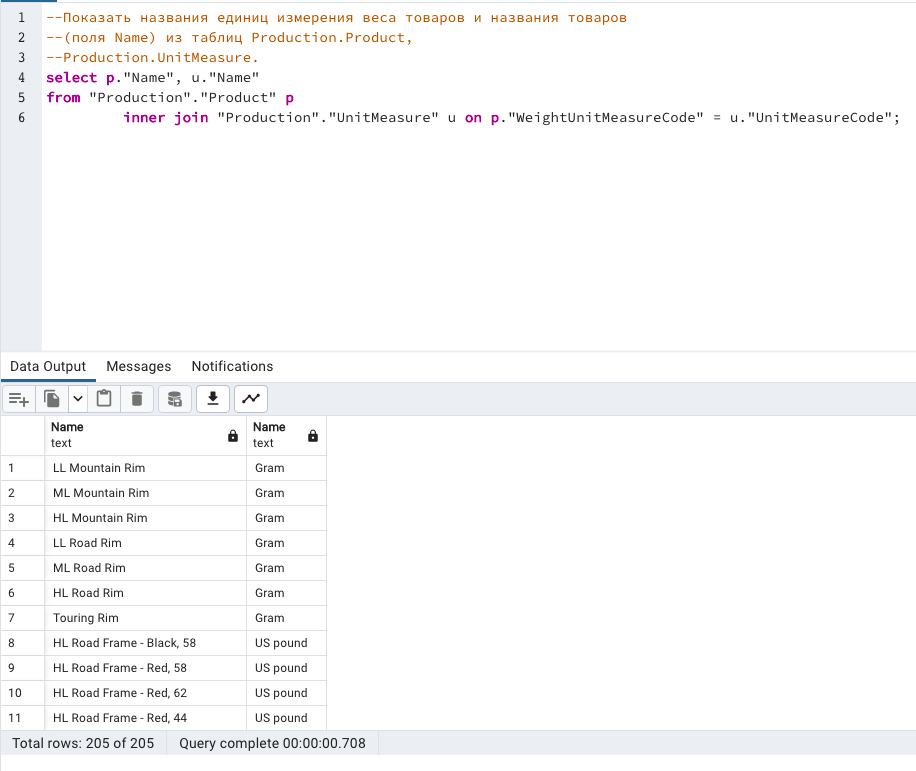


Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

Показать список продуктов (поле Name), в котором указано, есть дата окончания (EndDate) заказа или нет, из таблиц Production.Product, Production.WorkOrder, используя LEFT OUTER JOIN.

1. Текст скрипта:

--Показать список продуктов (поле Name), в котором указано, есть дата

--окончания (EndDate) заказа или нет, из таблиц

--Production.Product, Production.WorkOrder, используя

--LEFT OUTER JOIN.

select p."Name", w."EndDate"

from "Production"."Product" p

left outer join "Production"."WorkOrder" w on p."ProductID" = w."ProductID";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

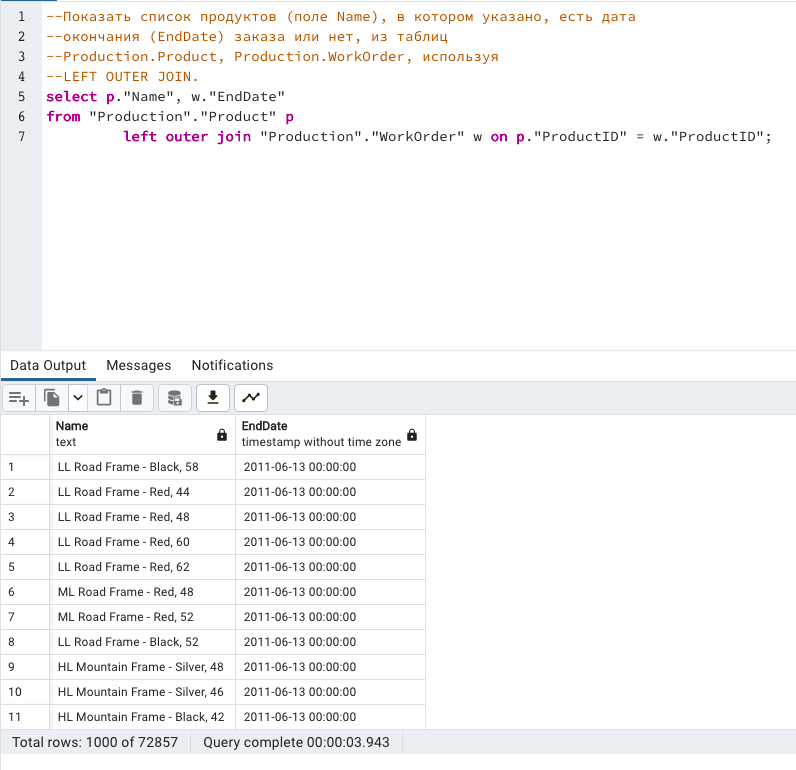


Рисунок 2 − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Показать список покупателей (поле AccountNumber из Sales.Customer), в котором указано, есть ли у покупателя код подтверждения кредитной карты (CreditCardApprovalCode) или нет, из таблиц Sales.SalesOrderHeader, Sales.Customer, используя RIGHT OUTER JOIN.

1. Текст скрипта:

--Показать список покупателей (поле AccountNumber из

--Sales.Customer), в котором указано, есть ли у покупателя код

--подтверждения кредитной карты (CreditCardApprovalCode)

--или нет, из таблиц Sales.SalesOrderHeader, Sales.Customer,

--используя RIGHT OUTER JOIN.

select c."AccountNumber", s."CreditCardApprovalCode"

from "Sales"."Customer" c

right outer join "Sales"."SalesOrderHeader" s on s."CustomerID" = c."CustomerID";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

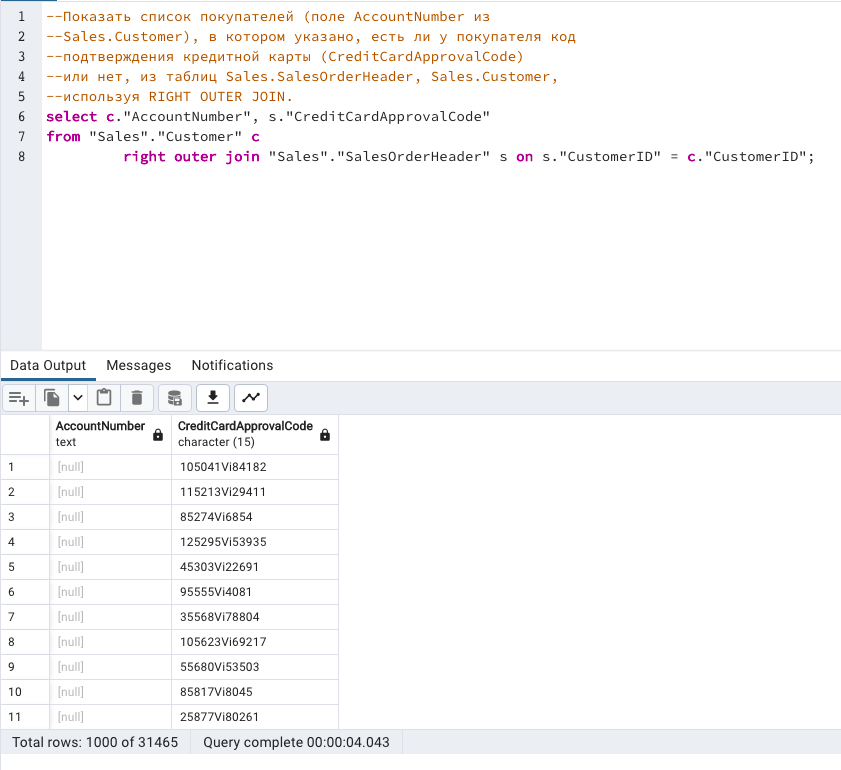


Рисунок 3 − Результат выполнения третьего задания

Задание 4

1. Формулировка задания:

Показать список типов спецпредложений (Type), которые имеют несколько дат окончания (EndDate), из таблицы Sales.SpecialOffer, используя SELF JOIN.

1. Текст скрипта:

--Показать список типов спецпредложений (Type), которые имеют

--несколько дат окончания (EndDate), из таблицы

--Sales.SpecialOffer, используя SELF JOIN.

select distinct s."Type"

from "Sales"."SpecialOffer" s

inner join "Sales"."SpecialOffer" s2 on s."Type" = s2."Type"

where s."EndDate" <> s2."EndDate";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

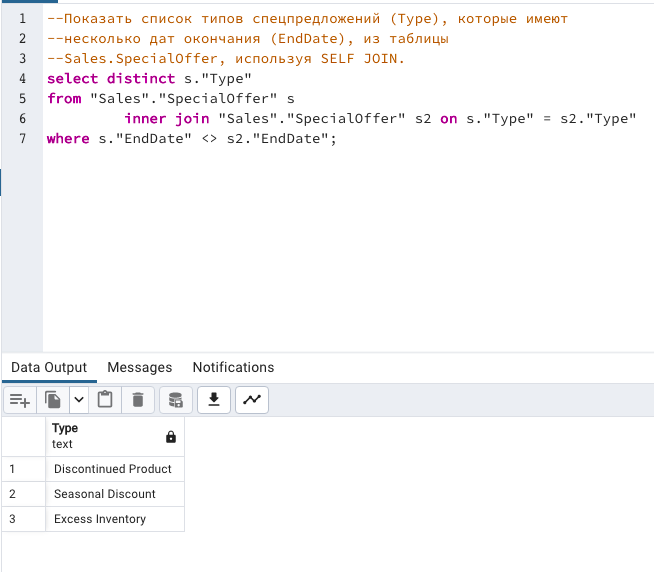


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

1. Формулировка задания:

Показать список дат окончания (EndDate) спецпредложения, к которым относятся несколько типов спецпредложений (Type), из таблицы Sales.SpecialOffer, используя SELF JOIN.

1. Текст скрипта:

--Показать список дат окончания (EndDate) спецпредложения, к

--которым относятся несколько типов спецпредложений (Type), из

--таблицы Sales.SpecialOffer, используя SELF JOIN.

select distinct s."EndDate"

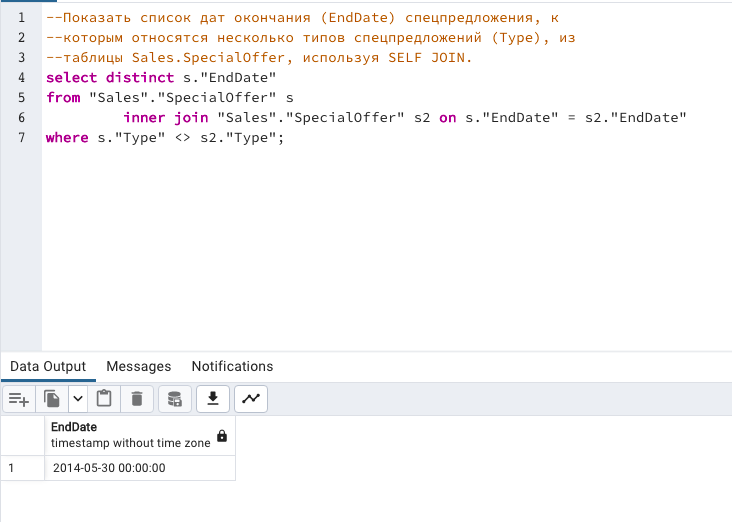
from "Sales"."SpecialOffer" s

inner join "Sales"."SpecialOffer" s2 on s."EndDate" = s2."EndDate"

where s."Type" <> s2."Type";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).



**Рисунок 5 − Результат выполнения пятого задания**

Задание 6

1. Формулировка задания:

Показать список продавцов (SalesPersonID из Sales.SalesOrderHeader), которые имеют нескольких покупателей, CustomerID которых заканчивается на 5, используя SELF JOIN.

1. Текст скрипта:

--Показать список продавцов (SalesPersonID из

--Sales.SalesOrderHeader), которые имеют нескольких

--покупателей, CustomerID которых заканчивается на 5, используя

--SELF JOIN.

select distinct s."SalesPersonID"

from "Sales"."SalesOrderHeader" s

inner join "Sales"."SalesOrderHeader" s2 on s."SalesPersonID" = s2."SalesPersonID"

where s."CustomerID" <> s2."CustomerID"

and s."CustomerID" % 5 = 0 and s."CustomerID" % 10 <> 0 and s2."CustomerID" % 5 = 0 and s2."CustomerID" % 10 <> 0;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

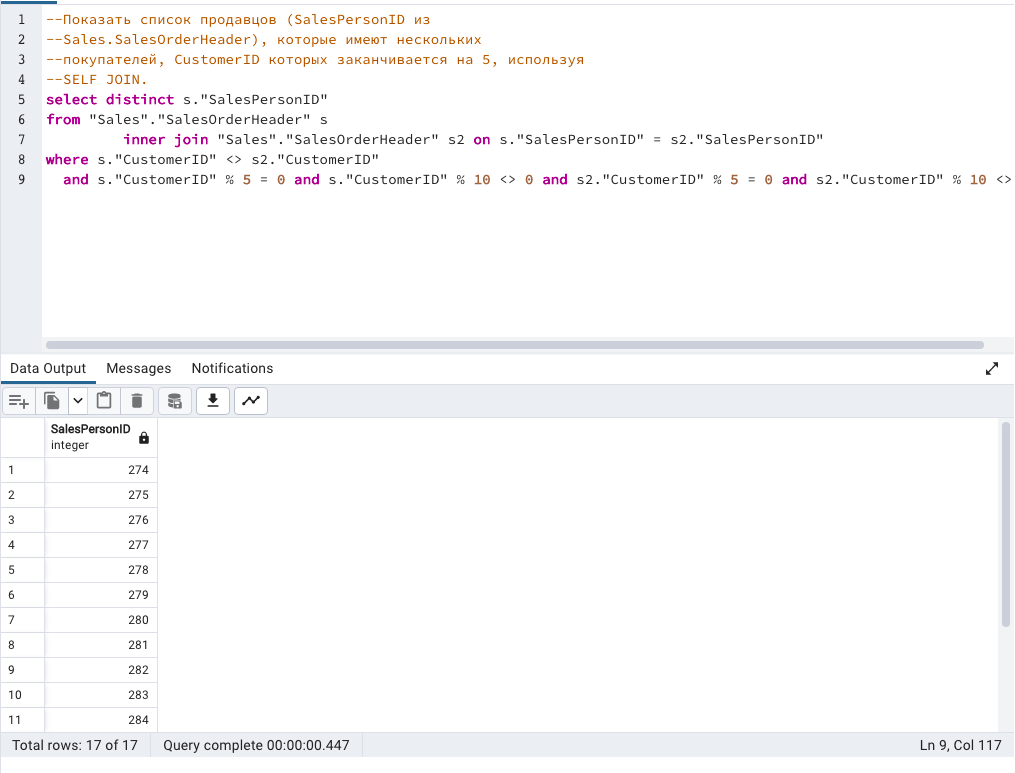


Рисунок 6 − Результат выполнения шестого задания

Задание 7

1. Формулировка задания:

Показать комбинированный список таблиц Purchasing.PurchaseOrderHeader по полям PurchaseOrderID, SubTotal, Purchasing.PurchaseOrderDetail по полям PurchaseOrderDetailID, UnitPrice, используя UNION.

1. Текст скрипта:

--Показать комбинированный список таблиц

--Purchasing.PurchaseOrderHeader по полям

--PurchaseOrderID, SubTotal,

--Purchasing.PurchaseOrderDetail по полям

--PurchaseOrderDetailID, UnitPrice, используя UNION.

select p."PurchaseOrderID", p."SubTotal"

from "Purchasing"."PurchaseOrderHeader" p

union

select p2."PurchaseOrderDetailID", p2."UnitPrice"

from "Purchasing"."PurchaseOrderDetail" p2;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

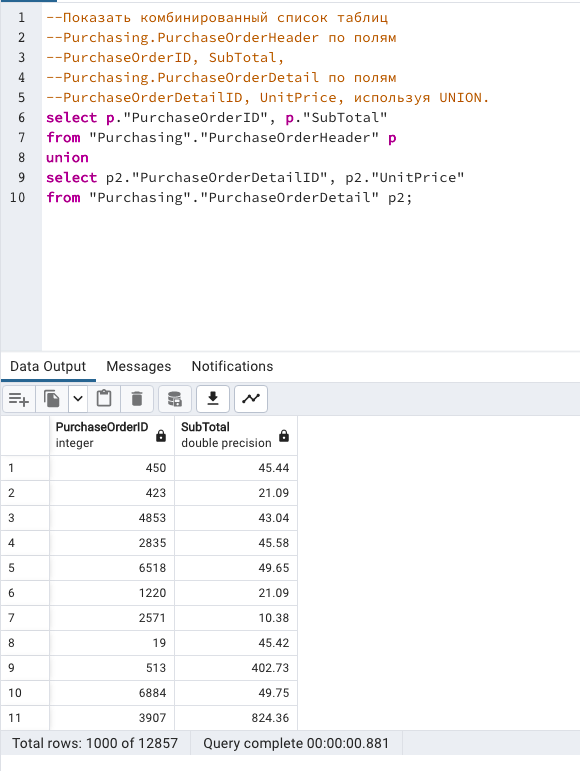


Рисунок 7 − Результат выполнения седьмого задания

Задание 8

1. Формулировка задания:

Показать список BusinessEntityID, которые содержатся в таблице Person.Person, но не содержатся в таблице Sales.PersonCreditCard.

1. Текст скрипта:

--Показать список BusinessEntityID, которые содержатся в

--таблице Person.Person, но не содержатся в таблице

--Sales.PersonCreditCard.

select p."BusinessEntityID"

from "Person"."Person" p

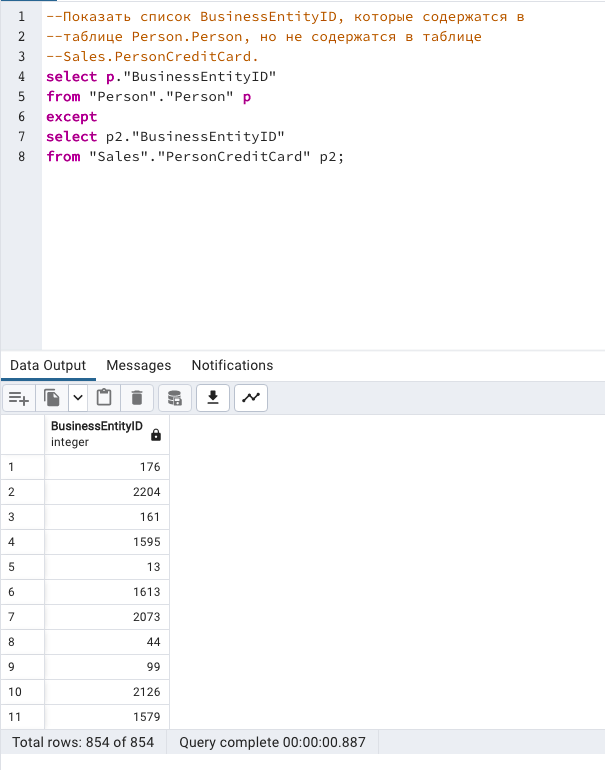
except

select p2."BusinessEntityID"

from "Sales"."PersonCreditCard" p2;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).



**Рисунок 8 − Результат выполнения седьмого задания**

Задание 9

1. Формулировка задания:

Ограничить результирующий набор, полученный в п.8.

1. Текст скрипта:

--Ограничить результирующий набор, полученный в п.8.

select p."BusinessEntityID"

from "Person"."Person" p

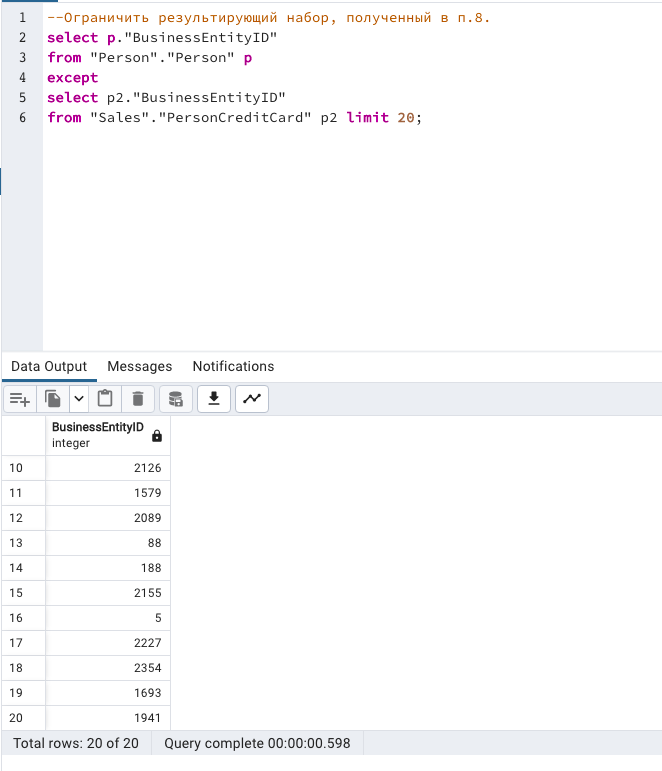
except

select p2."BusinessEntityID"

from "Sales"."PersonCreditCard" p2 limit 20;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).



**Рисунок 9 − Результат выполнения седьмого задания**

Задание 10

1. Формулировка задания:

Пропустить 15 строк из результирующего набора, полученного в п.9.

1. Текст скрипта:

--Пропустить 15 строк из результирующего набора, полученного в п.9.

select p."BusinessEntityID"

from "Person"."Person" p

except

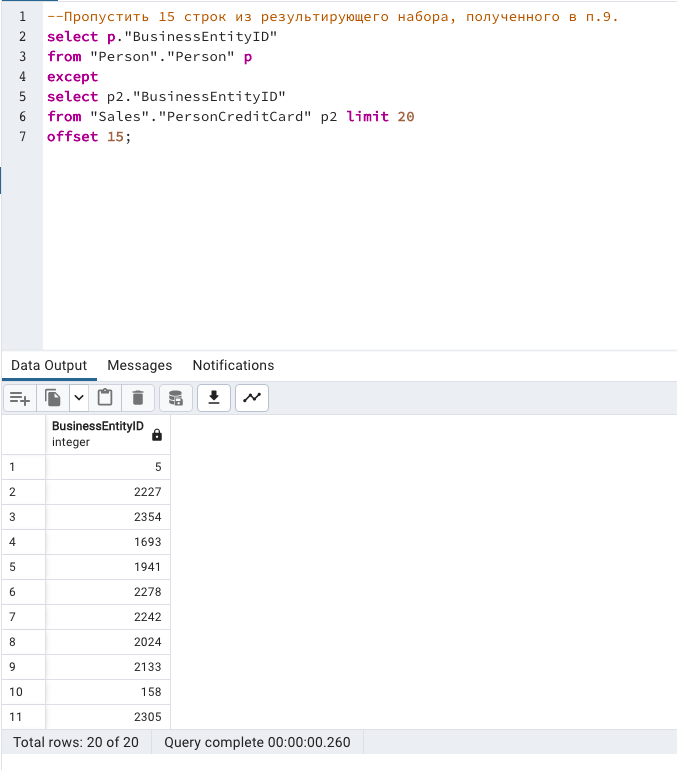
select p2."BusinessEntityID"

from "Sales"."PersonCreditCard" p2 limit 20

offset 15;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10).



**Рисунок 10 − Результат выполнения седьмого задания**

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

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

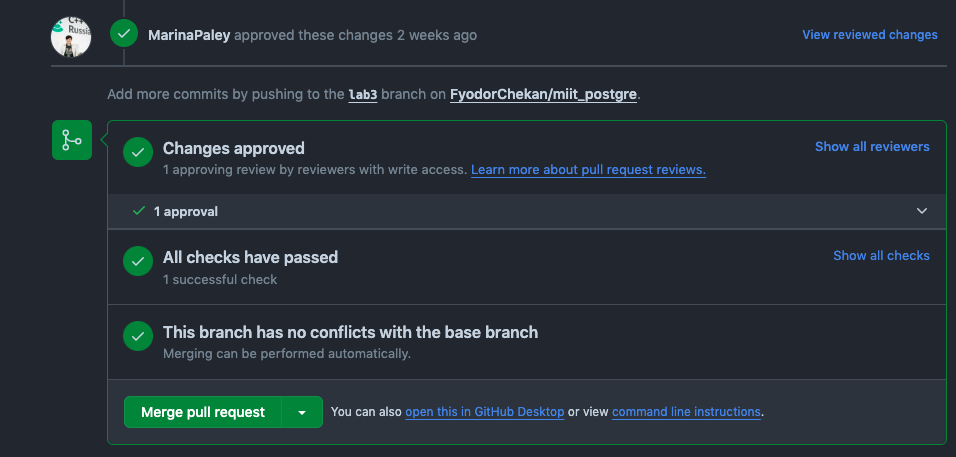


Рисунок 11 − Подтверждение о правильно выполненной работе