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 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 адреса электронной почты (EmailAddress) и поля PersonType, FirstName, LastName из таблиц Person.Person, Person.EmailAddress.

1. Текст скрипта:

SELECT

"EmailAddress",

P."PersonType",

P."FirstName",

P."LastName"

FROM

"Person"."Person" AS P

JOIN

"Person"."EmailAddress" AS E ON P."BusinessEntityID" = E."BusinessEntityID";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

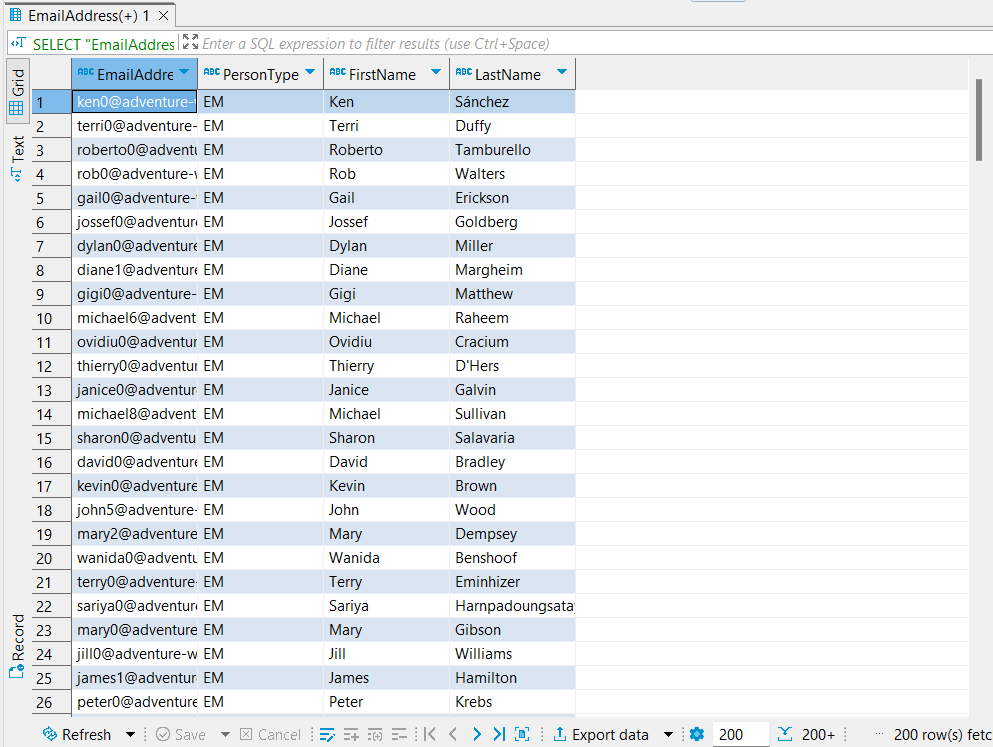


Рисунок 1 − Результат выполнения первого задания

Задание 2

1. Формулировка задания:

Показать список продуктов (поле Name), в котором указано, есть ли у продукта действительная цена (ActualCost) или нет, из таблиц Production.Product, Production.WorkOrderRouting, используя LEFT OUTER JOIN.

1. Текст скрипта:

SELECT

P."Name" AS ProductName,

CASE

WHEN WO."ActualCost" IS NOT NULL THEN 'Есть действительная цена'

ELSE 'Нет действительной цены'

END AS CostStatus

FROM

"Production"."Product" AS P

LEFT OUTER JOIN

"Production"."WorkOrderRouting" AS WO ON P."ProductID" = WO."ProductID";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

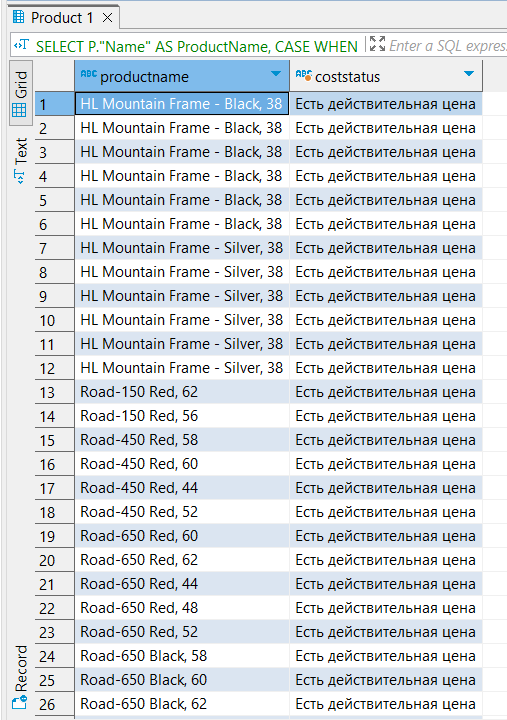


Рисунок 2 − Результат выполнения второго задания

Задание 3

1. Формулировка задания:

Показать список продуктов (поле Name), в котором указано, есть ли у продукта подкатегория или нет, из таблиц Production.ProductSubcategory, Production.Product, используя RIGHT OUTER JOIN.

1. Текст скрипта:

SELECT

P."Name" AS ProductName,

CASE

WHEN PS."ProductSubcategoryID" IS NOT NULL THEN 'Есть подкатегория'

ELSE 'Нет подкатегории'

END AS SubcategoryStatus

FROM

"Production"."Product" AS P

RIGHT OUTER JOIN

"Production"."ProductSubcategory" AS PS ON P."ProductSubcategoryID" = PS."ProductSubcategoryID";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

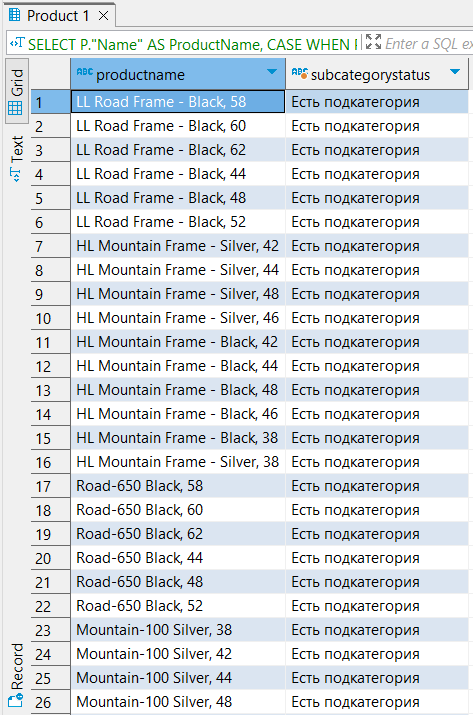


Рисунок 3 − Результат выполнения третьего задания

Задание 4

1. Формулировка задания:

Показать список подкатегорий продуктов (ProductSubcategoryID), которые имеют несколько нормативных цен (StandardCost), из таблицы Production.Product, используя SELF JOIN.

1. Текст скрипта:

SELECT

DISTINCT P1."ProductSubcategoryID",

COUNT(P1."StandardCost") AS NumberOfStandardCost

FROM

"Production"."Product" AS P1

INNER JOIN

"Production"."Product" AS P2 ON P1."ProductSubcategoryID" = P2."ProductSubcategoryID"

WHERE

P1."StandardCost" <> P2."StandardCost"

GROUP BY

P1."ProductSubcategoryID"

HAVING

COUNT(P1."StandardCost") > 1;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

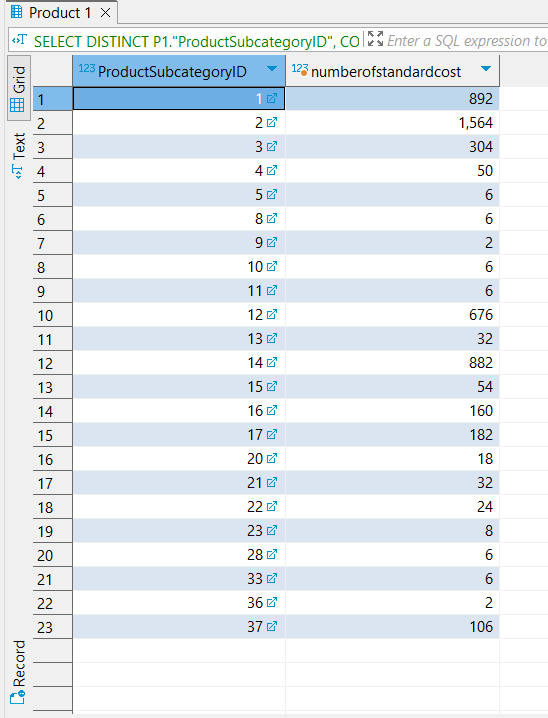


Рисунок 4 − Результат выполнения четвертого задания

Задание 5

1. Формулировка задания:

Показать список нормативных цен (StandardCost), которые имеют несколько подкатегорий продуктов (ProductSubcategoryID), из таблицы Production.Product, используя SELF JOIN.

1. Текст скрипта:

SELECT

DISTINCT P1."StandardCost",

COUNT(P1."ProductSubcategoryID") AS NumberOfProductSubcategories

FROM

"Production"."Product" AS P1

INNER JOIN

"Production"."Product" AS P2 ON P1."StandardCost" = P2."StandardCost"

WHERE

P1."ProductSubcategoryID" <> P2."ProductSubcategoryID"

GROUP BY

P1."StandardCost"

HAVING

COUNT(P1."ProductSubcategoryID") > 1;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

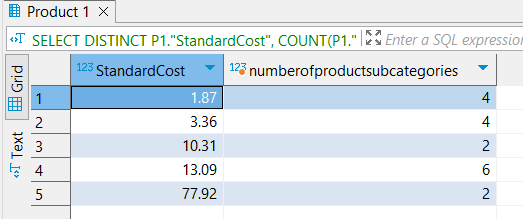


Рисунок 5 − Результат выполнения пятого задания

Задание 6

1. Формулировка задания:

Показать список типов спецпредложений (Type), которые имеют несколько значений минимального количества (MinQty) не меньше 15, из таблицы Sales.SpecialOffer, используя SELF JOIN.

1. Текст скрипта:

SELECT

DISTINCT S1."Type",

COUNT(S1."MinQty") AS NumberOfMinQtys

FROM

"Sales"."SpecialOffer" AS S1

INNER JOIN

"Sales"."SpecialOffer" AS S2 ON S1."Type" = S2."Type"

WHERE

S1."MinQty" >= 15 AND S1."MinQty" <> S2."MinQty"

GROUP BY

S1."Type"

HAVING

COUNT(S1."MinQty") > 1;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

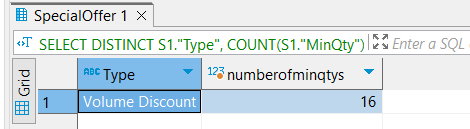


Рисунок 6 − Результат выполнения шестого задания

Задание 7

1. Формулировка задания:

Показать комбинированный список таблиц Person.BusinessEntityAddress, Person.BusinessEntityContact, используя UNION.

1. Текст скрипта:

SELECT \* FROM "Person"."BusinessEntityAddress"

UNION

SELECT \* FROM "Person"."BusinessEntityContact";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

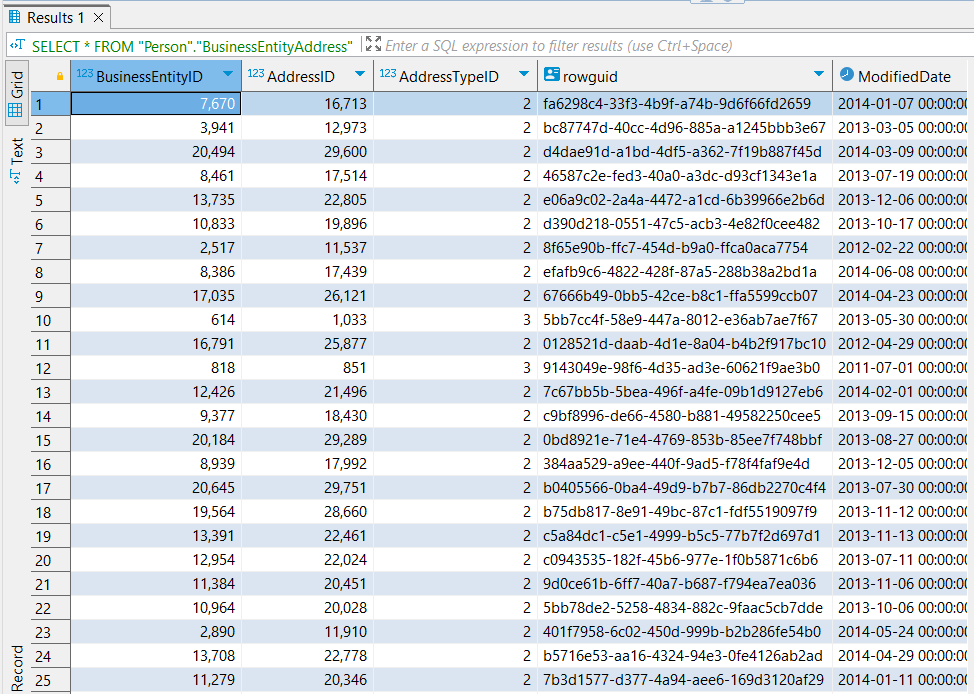


Рисунок 7 − Результат выполнения седьмого задания

Задание 8

1. Формулировка задания:

Показать список BusinessEntityID, которые содержатся в таблице Sales.SalesPerson, но не содержатся в таблице Sales.Store.

1. Текст скрипта:

SELECT "BusinessEntityID"

FROM "Sales"."SalesPerson"

EXCEPT

SELECT "BusinessEntityID"

FROM "Sales"."Store";

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

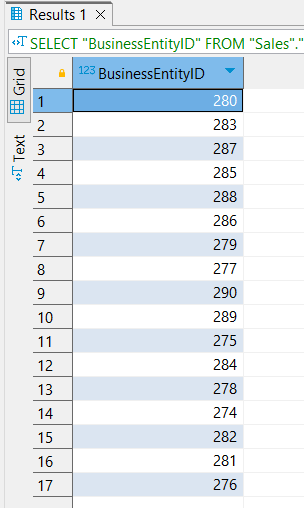


Рисунок 8 − Результат выполнения восьмого задания

Задание 9

1. Формулировка задания:

Ограничить результирующий набор, полученный в п.8.

1. Текст скрипта:

SELECT sp."BusinessEntityID"

FROM "Sales"."SalesPerson" sp

EXCEPT

SELECT ss."BusinessEntityID"

FROM "Sales"."Store" ss

LIMIT 10;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).

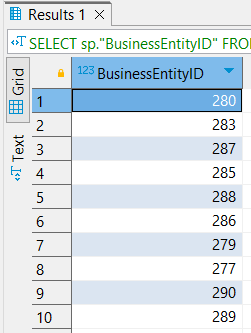


Рисунок 9 − Результат выполнения девятого задания

Задание 10

1. Формулировка задания:

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

1. Текст скрипта:

SELECT sp."BusinessEntityID"

FROM "Sales"."SalesPerson" sp

EXCEPT

SELECT ss."BusinessEntityID"

FROM "Sales"."Store" ss

OFFSET 10;

1. Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10).

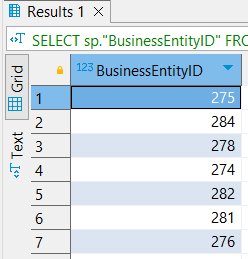


Рисунок 10 − Результат выполнения десятого задания

## 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