|  |
| --- |
| МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА» (РУТ (МИИТ))  ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Кафедра «Управление и защита информации» |
|  |
| **ЛАБОРАТОРНАЯ РАБОТА №2**  **по дисциплине «Основы построения защищенных баз данных»**  Вариант №2 |
|  |
|  |
|  |
| Выполнил: ст. гр. ТКИ – 441  Галыба Л.Д. |
| Проверил: к.т.н., доц.,  Васильева М.А. |
|  |
|  |
|  |
| Москва 2024 |

**Оглавление**

[Цель работы 3](#_Toc161314466)

[Выполнение работы 4](#_Toc161314467)

[Задание 1 4](#_Toc161314468)

[Задание 2 4](#_Toc161314469)

[Задание 3 5](#_Toc161314470)

[Задание 4 6](#_Toc161314471)

[Задание 5 6](#_Toc161314472)

[Задание 6 7](#_Toc161314473)

[Задание 7 8](#_Toc161314474)

[Результат проверки преподавателем 10](#_Toc161314475)

[Вывод 11](#_Toc161314476)

Цель работы

Изучить операторы SQL диалекта Postgres Pro, необходимые для группировки и обобщения данных. Научиться создавать запросы с агрегатными функциями. Подготавливать отчет, выполненный согласно требованиям ГОСТ 7.32.

Выполнение работы

Задание 1

Найти минимальную цену товара (ListPrice) из таблицы Production.Product. Не учитывать нулевую цену.

**select**

**min**("ListPrice")

**from** "Production"."Product" *p*

**where**

"ListPrice" > 0

|  |
| --- |
|  |
| 1. – Выполнение скрипта №1 |

Задание 2

Найти средний вес (Weight) из таблицы Production.Product, не учитывать пустые значения.

**select**

**avg**("Weight")

**from** "Production"."Product" *p*

**where**

"Weight" **is** **not** **null**

|  |
| --- |
|  |
| 1. – Выполнение скрипта №2 |

Задание 3

Для каждого пола (Gender) найти суммарное количество часов отпуска из таблицы (поставить фильтр: работник был принят до 2008 года) HumanResources.Employee, не учитывать пустые значения.

**select**

"Gender"

, **sum**("VacationHours") **as** *"Total"*

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

**where**

"Gender" **is** **not** **null**

**and** "VacationHours" **is** **not** **null**

**and** "HireDate" < **'2008-01-01'**

**group** **by**("Gender")

|  |
| --- |
|  |
| 1. – Выполнение скрипта №3 |

Задание 4

Вывести пол, где среднее количество часов выходных больше 50 из таблицы HumanResources.Employee.

**select**

"Gender"

, **avg**("VacationHours") **as** *"Average"*

**from**

"HumanResources"."Employee"

**group** **by** ("Gender")

**having** **avg**("VacationHours") > 50

|  |
| --- |
|  |
| 1. – Выполнение скрипта №4 |

Задание 5

Вывести BusinessEntityID, LastReceiptDate и максимальное значение LastReceiptCost для BusinessEntityID с разными LastReceiptDate, предусмотреть вывод общей суммы для всех LastReceiptCost у различных BusinessEntityID из таблицы Purchasing.ProductVendor. (Использовать ROOLUP).

**select**

"BusinessEntityID"

, "LastReceiptDate"

, **max**("LastReceiptCost") **as** *"MaxLastReceiptCost"*

, **sum**("LastReceiptCost") **as** *"TotalCost"*

**from**

"Purchasing"."ProductVendor" *pv*

**group** **by** **rollup**("LastReceiptDate", "BusinessEntityID")

|  |
| --- |
|  |
| 1. – Выполнение скрипта №5 |

Задание 6

Изменить запрос п.5 использовать CUBE. Отделить строки, созданные с помощью агрегатных функций от строк из фактической таблицы.

**select**

"BusinessEntityID"

, "LastReceiptDate"

, **max**("LastReceiptCost") **as** *"MaxLastReceiptCost"*

, **sum**("LastReceiptCost") **as** *"TotalCost"*,

**grouping**("BusinessEntityID", "LastReceiptDate")

**from**

"Purchasing"."ProductVendor" *pv*

**group** **by** **cube**("LastReceiptDate", "BusinessEntityID")

|  |
| --- |
|  |
| 1. – Выполнение скрипта №6 |

Задание 7

Изменить запрос п.5 использовать GROUPING SETS. Отделить строки, созданные с помощью агрегатных функций от строк из фактической таблицы.

**select**

"BusinessEntityID"

, "LastReceiptDate"

, **max**("LastReceiptCost") **as** *"MaxLastReceiptCost"*

, **sum**("LastReceiptCost") **as** *"TotalCost"*

**from**

"Purchasing"."ProductVendor" *pv*

**group** **by** **grouping** **sets**("LastReceiptDate", "BusinessEntityID")

|  |
| --- |
|  |
| 1. – Выполнение скрипта №7 |

Результат проверки преподавателем

|  |
| --- |
|  |
| 1. – Подтверждение от преподавателя |

Вывод

В результате выполнения лабораторной работы были изучены операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для группировки и обобщения данных. Получен навык создания запросов с агрегатными функциями.