|  |  |
| --- | --- |
| 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 №1 | |
|  |  |
|  |  |
| **«Основы построения защищенных баз данных»** | |
|  |  |
|  | Выполнил: ст. гр. ТКИ-441 |
|  | Порхун Д.Д. |
|  | Вариант №13 |
|  | Проверил: доц. Васильева М. А. |
|  |  |
|  |  |
|  | |
| Москва – 2024г. | |
|  | |

[Введение 2](#_Toc159265676)

[1. Цель работы 2](#_Toc159265677)

[2. Описание задачи 2](#_Toc159265678)

[3. Экспериментальная часть 3](#_Toc159265679)

[Вывод 11](#_Toc159265680)

Введение

1. Цель работы

Выполнить вторую лабораторную работу, Изучить операторы SQL диалекта PostgresPro, необходимые для группировки и обобщения данных. Научиться создавать запросы с агрегатными[[1]](#footnote-1) функциями.

1. Описание задачи

1. Найти среднюю стоимость (StandardCost) из таблицы ProductCostHistory.

2. Найти пароли, начинающиеся с I или i, из таблицы Person.Password.

3. Найти сумму часов выходных для разных полов из таблицы HumanResources.Employee.

4. Показать города, основываюсь на таблице Person.Address, где количество заказов было больше 10 (из выборки исключить пустые поля).

5. Вывести ProductID из таблицы Production.ProductListPriceHistory у которых максимальная цена (ListPrice) больше 24. Добавить столбец с рангом, определяя его порядок в зависимости от количества строк, содержащих максимальную цену. (Использовать RANK).

1. Экспериментальная часть

Листинг задачи 1:

|  |
| --- |
| -- 1. Найти среднюю стоимость (StandardCost) из таблицы ProductCostHistory.  **select** **avg**("StandardCost") **from** "Production"."ProductCostHistory"; |
|  |
| 1. Визуализация выполненного запроса к задаче 1 |

Листинг задачи 2:

|  |
| --- |
| -- 2. Найти пароли, начинающиеся с I или i, из таблицы Person.Password.  **select** "PasswordSalt" **from** "Person"."Password" **where** "PasswordSalt" **LIKE** **'I%'** **or** "PasswordSalt" **LIKE** **'i%'** |
| https://sun9-45.userapi.com/impg/1vVkdDqJlnHJrHsD60zbQ_s6WCFodshKlcNgzA/oOvzixXdmBw.jpg?size=193x276&quality=96&sign=df92d0d79cde5c267261ebfebcb8b6ff&type=album |
| 1. Визуализация выполненного запроса к задаче 2 |

Листинг задачи 3:

|  |
| --- |
| -- 3. Найти сумму часов выходных для разных полов из таблицы HumanResources.Employee.  **SELECT** **SUM**(**CASE** **WHEN** "Gender" = **'M'** **THEN** "SickLeaveHours" **ELSE** 0 **END**) **AS** *"SumTimeMale"*, **SUM**(**CASE** **WHEN** "Gender" = **'F'** **THEN** "SickLeaveHours" **ELSE** 0 **END**) **AS** *"SumTimeFemale"*  **FROM** "HumanResources"."Employee"; |
|  |
| 1. Визуализация выполненного запроса к задаче 3 |

Листинг задачи 4:

|  |
| --- |
| -- 4. Показать города, основываюсь на таблице Person.Address, где количество заказов было больше 10 (из выборки исключить пустые поля).  **SELECT** "City" , **COUNT**("StateProvinceID") **FROM** "Person"."Address"  **WHERE** "StateProvinceID" **IS** **NOT** **NULL**  **GROUP** **by** "City"  **having** **COUNT**("StateProvinceID") > 10; |
|  |
| 1. Визуализация выполненного запроса к задаче 4 |

Листинг задачи 5:

|  |
| --- |
| -- 5. Вывести ProductID из таблицы Production.ProductListPriceHistory у которых максимальная цена (ListPrice) больше 24.  -- Добавить столбец с рангом, определяя его порядок в зависимости от количества строк, содержащих максимальную цену. (Использовать RANK).  **select** "ProductID", **RANK**() **OVER**(**ORDER** **BY** **count**("ProductID") **DESC**), **count**("ProductID") **as** *"Count"*  **from** "Production"."ProductListPriceHistory" **where** "ListPrice" > 24 **group** **by** "ProductID"; |
| https://sun9-2.userapi.com/impg/WrzQQuvs2pIVg5WgeuUxgU0B6dmnGOeoXnSMCQ/ADzpBchrguE.jpg?size=357x354&quality=96&sign=766d90cb82ad95c8c48996835d8fe3a1&type=album |
| 1. Визуализация выполненного запроса к задаче 5 |

Вывод

По ходу выполнения лабораторной работы, были изучены операторы SQL диалекта PostgresPro, необходимые для группировки и обобщения данных. Научиться создавать запросы с агрегатными функциями.

1. статистические функции [↑](#footnote-ref-1)