|  |
| --- |
| 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 №3**  **по дисциплине «Основы построения защищенных баз данных»**  Вариант №14 |
|  |
|  |
|  |
| Выполнил: ст. гр. ТКИ – 441  Поткин Д.А. |
| Проверил: к.т.н., доц.,  Васильева М.А. |
|  |
|  |
|  |
| Москва 2024 |

**Оглавление**

[Цель работы 3](#_Toc161776117)

[Выполнение работы 4](#_Toc161776118)

[Задание 1 4](#_Toc161776119)

[Задание 2 4](#_Toc161776120)

[Задание 3 5](#_Toc161776121)

[Задание 4 6](#_Toc161776122)

[Задание 5 7](#_Toc161776123)

[Задание 6 8](#_Toc161776124)

[Задание 7 9](#_Toc161776125)

[Задание 8 10](#_Toc161776126)

[Задание 9 11](#_Toc161776127)

[Задание 10 12](#_Toc161776128)

[Результат проверки преподавателем 14](#_Toc161776129)

[Вывод 15](#_Toc161776130)

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.

Рабочее задание:

1. Показать названия подкатегорий и соответствующие им названия категорий товаров (поля Name) из таблиц. ProductSubcategory, Production.ProductCategory.
2. Показать список складов (поле Name), в котором указано, есть на складе товар с некоторым ID или нет, из таблиц Production.Location, Production.WorkOrderRouting, используя LEFT OUTER JOIN.
3. Показать список людей (поля FirstName, LastName), в котором указано, есть ли у человека часы больничного (SickLeaveHours) или нет, из таблиц HumanResources.Employee, Person.Person, используя RIGHT OUTER JOIN.
4. Показать список покупателей (CustomerID), которые имеют несколько стоимостей доставки (Freight), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.
5. Показать список стоимостей доставки (Freight), к которым относятся несколько покупателей (CustomerID), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.
6. Показать список имен (FirstName) начинающихся на «N», которые имеют несколько фамилий (LastName) из таблицы Person.Person, используя SELF JOIN.
7. Показать комбинированный список таблиц Production.Product по полям ProductID, ListPrice, Purchasing.PurchaseOrderDetail по полям ProductID, UnitPrice, используя UNION.
8. Показать список ProductID, которые содержатся и в таблице Production.Product, и в таблице Production.TransactionHistoryArchive.
9. Ограничить результирующий набор, полученный в п.8.
10. Пропустить 15 строк из результирующего набора, полученного в п.9.

Выполнение работы

Задание 1

Показать названия подкатегорий и соответствующие им названия категорий товаров (поля Name) из таблиц. ProductSubcategory, Production.ProductCategory.

**select** "Production"."ProductSubcategory"."Name" **as** *"SubcategoryName"*,

"Production"."ProductCategory"."Name" **as** *"CategoryName"*

**from** "Production"."ProductSubcategory"

**join** "Production"."ProductCategory" **on** "Production"."ProductSubcategory"."ProductCategoryID" = "Production"."ProductCategory"."ProductCategoryID";

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

Задание 2

Показать список складов (поле Name), в котором указано, есть на складе товар с некоторым ID или нет, из таблиц Production.Location, Production.WorkOrderRouting, используя LEFT OUTER JOIN.

**select** "Production"."Location"."Name",

"Production"."WorkOrderRouting"."LocationID" **as** *"AvailableProduct"*

**from** "Production"."Location"

**left** **outer** **join** "Production"."WorkOrderRouting" **on** "Production"."Location"."LocationID" = "Production"."WorkOrderRouting"."LocationID";

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

Задание 3

Показать список людей (поля FirstName, LastName), в котором указано, есть ли у человека часы больничного (SickLeaveHours) или нет, из таблиц HumanResources.Employee, Person.Person, используя RIGHT OUTER JOIN.

**select** "Person"."Person"."FirstName",

"Person"."Person"."LastName",

"HumanResources"."Employee"."SickLeaveHours"

**from** "Person"."Person"

**right** **outer** **join** "HumanResources"."Employee" **on** "Person"."Person"."BusinessEntityID" = "HumanResources"."Employee"."BusinessEntityID" ;

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

Задание 4

Показать список покупателей (CustomerID), которые имеют несколько стоимостей доставки (Freight), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.

**select** **distinct** *so1*."CustomerID"

**from** "Sales"."SalesOrderHeader" *so1*

**inner** **join** "Sales"."SalesOrderHeader" *so2* **on** *so1*."CustomerID" = *so2*."CustomerID"

**where** *so1*."Freight" <> *so2*."Freight";

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

Задание 5

Показать список стоимостей доставки (Freight), к которым относятся несколько покупателей (CustomerID), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.

**select** *so1*."Freight"

**from** "Sales"."SalesOrderHeader" *so1*

**inner** **join** "Sales"."SalesOrderHeader" *so2* **on** *so1*."Freight" = *so2*."Freight" **and** *so1*."CustomerID" <> *so2*."CustomerID"

**group** **by** *so1*."Freight"

**having** **count**(**distinct** *so1*."CustomerID") > 1;

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

Задание 6

Показать список имен (FirstName) начинающихся на «N», которые имеют несколько фамилий (LastName) из таблицы Person.Person, используя SELF JOIN.

**select** *p1*."FirstName"

**from** "Person"."Person" *p1*

**inner** **join** "Person"."Person" *p2* **on** *p1*."FirstName" = *p2*."FirstName" **and** *p1*."LastName" <> *p2*."LastName"

**where** *p1*."FirstName" **like** **'N%'**

**group** **by** *p1*."FirstName"

**having** **count**(**distinct** *p1*."LastName") > 1;

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

Задание 7

Показать комбинированный список таблиц Production.Product по полям ProductID, ListPrice, Purchasing.PurchaseOrderDetail по полям ProductID, UnitPrice, используя UNION.

**select** *p*."ProductID",

*p*."ListPrice",

**null** **as** **"UnitPrice"**

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

**union**

**select** *pod*."ProductID",

**null** **as** **"ListPrice"**,

*pod*."UnitPrice"

**from** "Purchasing"."PurchaseOrderDetail" *pod*;

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

Задание 8

Показать список ProductID, которые содержатся и в таблице Production.Product, и в таблице Production.TransactionHistoryArchive.

**select** "ProductID"

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

**intersect**

**select** "ProductID"

**from** "Production"."TransactionHistoryArchive";

|  |
| --- |
|  |
| 1. – Выполнение скрипта №8 |

Задание 9

Ограничить результирующий набор, полученный в п.8.

**select** "ProductID"

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

**intersect**

**select** "ProductID"

**from** "Production"."TransactionHistoryArchive"

**order** **by** "ProductID"

**limit** 20;

|  |
| --- |
|  |
| 1. – Выполнение скрипта №9 |

Задание 10

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

**select** "ProductID"

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

**intersect**

**select** "ProductID"

**from** "Production"."TransactionHistoryArchive"

**order** **by** "ProductID"

**limit** 20

**offset** 15;

|  |
| --- |
|  |
| 1. – Выполнение скрипта №10 |

Результат проверки преподавателем

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

Вывод

В результате выполнения лабораторной работы были изучены операторы SQL диалекта PostgreSQL, необходимые для работы с данными, находящимися в разных таблицах. Получен навык создания запросов на соединение данных из множества таблиц.