Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**О Т Ч Ё Т**

по лабораторной работе №3

по дисциплине:

«Информационное обеспечение систем управления»

на тему:

«Соединение данных из множества таблиц»

Выполнил: ст. гр. ВУЦ-421

Полунин С.К.

Вариант 6

Проверил: доц., к.т.н.

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**

Формулировка задания:

Показать процент комиссии (CommissionPct) и поля FirstName, LastName из таблиц PersonPerson, Sales.SalesPerson.

Текст скрипта:

**select** *sp*."CommissionPct"

,*p*."FirstName"

,*p*."LastName"

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

**inner** **join** "Sales"."SalesPerson" *sp*

**on** *p*."BusinessEntityID" = *sp*."BusinessEntityID"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

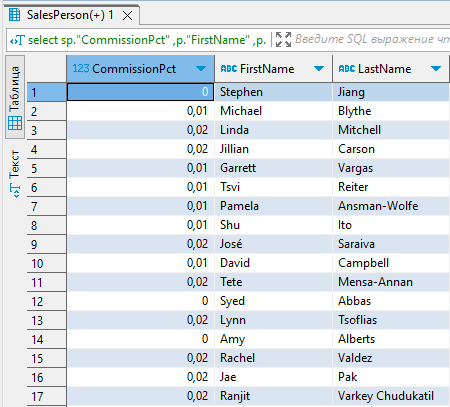


Рисунок 1 – Результат выполнения первого задания.

**Задание** **2**

Формулировка задания:

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

Текст скрипта:

**select** *p*."Name"

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

**left** **outer** **join** "Production"."ProductReview" *pr*

**on** *p*."ProductID" = *pr*."ProductID"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

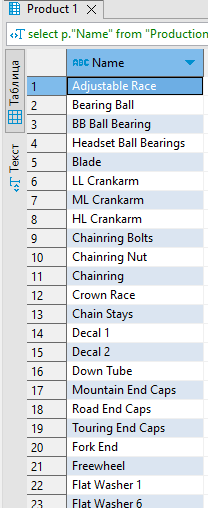


Рисунок 2 – Результат выполнения второго задания.

**Задание** **3**

Формулировка задания:

Показать список ID людей (поле BusinessEntityId), в котором указано, является человек вендором или нет, из таблиц Purchasing.Vendor, Person.BusinessEntity, используя RIGHT OUTER JOIN.

Текст скрипта:

**select** *pb*."BusinessEntityID"

,*pv*."BusinessEntityID"

**from** "Person"."BusinessEntity" *pb*

**right** **outer** **join** "Purchasing"."Vendor" *pv*

**on** *pb*."BusinessEntityID" = *pv*."BusinessEntityID"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

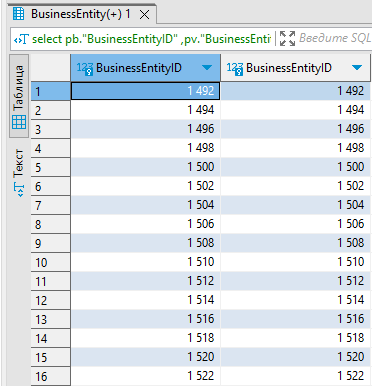


Рисунок 3 – Результат выполнения третьего задания.

**Задание** **4**

Формулировка задания:

Показать список продуктов (ProductID), которые имеют несколько дат выполнения заказа (DueDate), из таблицы Purchasing.PurchaseOrderDetail, используя SELF JOIN.

Текст скрипта:

**select** **distinct**

*p1*."ProductID"

**from** "Purchasing"."PurchaseOrderDetail" *p1*

**inner** **join** "Purchasing"."PurchaseOrderDetail" *p2*

**on** *p1*."ProductID" = *p2*."ProductID"

**where** *p1*."DueDate" <> *p2*."DueDate"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

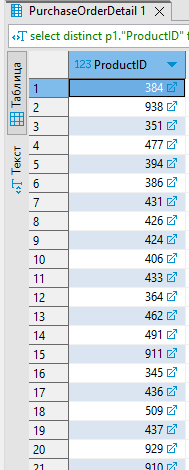


Рисунок 4 − Результат выполнения четвертого задания.

**Задание** **5**

Формулировка задания:

Показать список дат выполнения заказа (DueDate), к которым относятся несколько продуктов (ProductID), из таблицы Purchasing.PurchaseOrderDetail, используя SELF JOIN.

Текст скрипта:

**select** **distinct**

*p1*."DueDate"

**from** "Purchasing"."PurchaseOrderDetail" *p1*

**inner** **join** "Purchasing"."PurchaseOrderDetail" *p2*

**on** *p1*."DueDate" <> *p2*."DueDate"

**where** *p1*."ProductID" = *p2*."ProductID"

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).

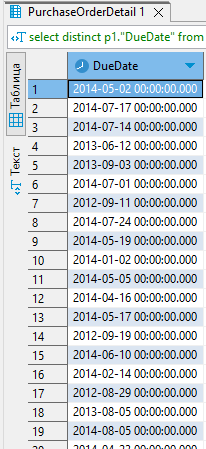


Рисунок 5 − Результат выполнения пятого задания.

**Задание** **6**

Формулировка задания:

Показать список покупателей (CustomerID), которые имеют несколько стоимостей доставки (Freight), из таблицы Sales.SalesOrderHeader, используя SELF JOIN. Значение стоимостей не менее 50.

Текст скрипта:

**select** **distinct** *s1*."CustomerID"

**from** "Sales"."SalesOrderHeader" *s1*

**inner** **join** "Sales"."SalesOrderHeader" *s2*

**on** *s1*."CustomerID" = *s2*."CustomerID"

**where** *s1*."Freight" <> *s2*."Freight" **and** *s1*."Freight" >= 50;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).

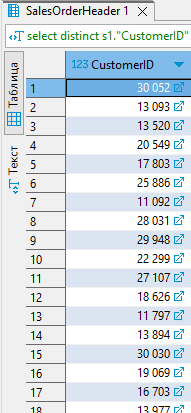


Рисунок 6 − Результат выполнения шестого задания.

**Задание** **7**

Формулировка задания:

Показать комбинированный список таблиц Person.EmailAddress, Person.BusinessEntityAddress по полям BusinessEntityID, rowguid, ModifiedDate, используя UNION.

Текст скрипта:

**select** "BusinessEntityID", "rowguid", "ModifiedDate"

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

**union**

**select** "BusinessEntityID", "rowguid", "ModifiedDate"

**from** "Person"."BusinessEntityAddress";

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

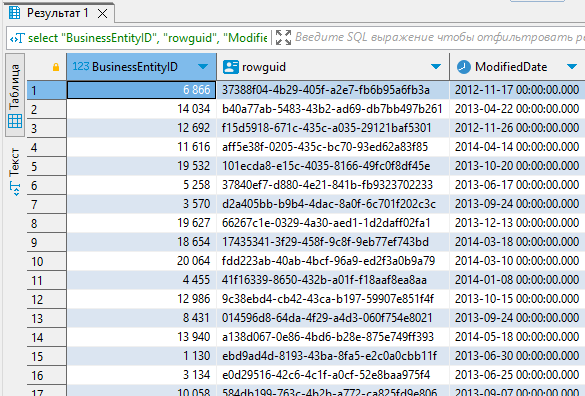


Рисунок 7 − Результат выполнения седьмого задания.

**Задание** **8**

Формулировка задания:

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

Текст скрипта:

**select** "ProductID"

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

**except**

**select** "ProductID"

**from** "Production"."ProductInventory";

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

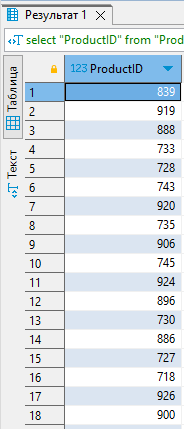


Рисунок 8 − Результат выполнения восьмого задания.

**Задание** **9**

Формулировка задания:

Ограничить результирующий набор, полученный в п.8.

Текст скрипта:

**select** "ProductID"

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

**limit** 7

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 9).

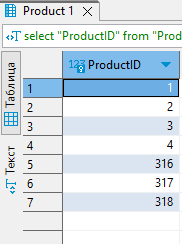


Рисунок 9 − Результат выполнения девятого задания.

**Задание** **10**

Формулировка задания:

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

Текст скрипта:

**select** **"ProductID"**

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

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

**offset** 15

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 10).

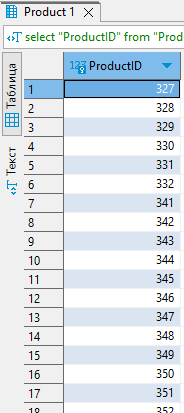


Рисунок 10 − Результат выполнения десятого задания.

Вывод: изучены операторы SQL диалекта PostgreSQL, необходимые для работы с данными, находящимися в разных таблицах. Научились создавать запросы на соединение данных из множества таблиц.