

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

«Информационное обеспечение систем управления»

Выполнил: ст. гр. ТУУ-411

Дудкин А.В.

Вариант №6

Проверил: доц. Васильева М. А.

Москва 2023

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ	2
2. ОПИСАНИЕ ЗАДАЧИ	2
3. ЭКСПЕРЕМЕНТАЛЬНАЯ ЧАСТЬ	3
4. ВЫВОД	7

1. ЦЕЛЬ РАБОТЫ

Изучить операторы, необходимые для работы с данными, находящимися в разных таблицах. Научиться создавать запросы на соединение данных из множества таблиц. Работа производится с учебной базой данной AdventureWorks.

2. ОПИСАНИЕ ЗАДАЧИ

1 Показать процент комиссии (ComissionPct) и поля FirstName, LastName из таблиц Person.Person, Sales.SalesPerson.

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

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

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

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

*6 Показать список покупателей (CustomerID), которые имеют несколько стоимостей доставки (Freight), из таблицы Sales.SalesOrderHeader, используя SELF JOIN. Значение стоимостей не менее 50 (из-за версии Postfres 9.5 нет доступа к таблице Sales.SalesOrdersHeader) Поэтому пункт 6 взят из 23 варианта:
6 Показать товары из таблицы Production.Product, имеющие такой же вес как у товара с названием «LL Road Frame - Black, 62», используя SELF JOIN.

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

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

3. ЭКСПЕРЕМЕНТАЛЬНАЯ ЧАСТЬ

На рисунке 1 реализация задачи 1:

The screenshot displays the SQL Server Enterprise Manager interface. At the top, a query window shows the following SQL code:

```
--1. Показать процент комиссии (ComissionPct) и поля FirstName, LastName из таблиц PersonPerson, Sales.SalesPerson.  
select p."FirstName"  
      , p."LastName"  
      , sp."ComissionPct"  
from "Person"."Person" p  
--объединения строк (join on) из разных таблиц, сработает в случае строки будут объединены только в том случае, если значения  
--столбца BusinessEntityID в таблице Person.Person равно значению столбца BusinessEntityID в таблице Sales.SalesPerson  
join "Sales"."SalesPerson" sp on p."BusinessEntityID" = sp."BusinessEntityID";
```

Below the query window, the results of the query are displayed in a table. The table has four columns: 'First Name', 'Last Name', and 'CommissionPct'. The results are as follows:

	First Name	Last Name	CommissionPct
1	Stephen	Jiang	0
2	Michael	Blythe	0,01
3	Linda	Mitchell	0,02
4	Jillian	Carson	0,02
5	Garrett	Vargas	0,01
6	Tsvi	Reiter	0,01
7	Pamela	Ansman-Wolfe	0,01
8	Shu	Ito	0,01
9	José	Saraiva	0,02
10	David	Campbell	0,01
11	Tete	Mensa-Annan	0,02
12	Syed	Abbas	0
13	Lynn	Tsoflias	0,02
14	Amy	Alberts	0
15	Rachel	Valdez	0,02
16	Jae	Pak	0,02
17	Ranjit	Varkey Chudukatil	0,02

On the right side of the interface, a 'Значение' (Value) pane shows the value 'Stephen' for the first row of the results.

Рисунок 1

На рисунке 2 реализация задачи 2:

--2 Показать список продуктов (поле Name), в котором указано, есть ли у продукта рейтинг или нет, из таблиц Production.Product, Production.ProductReview, используя LEFT OUTER JOIN.

```

select p."Name"
, coalesce (cast(pr."ProductID" as VARCHAR (50)), 'Нет рейтинга') as Рейтинг_продукта
from "Production"."Product" p

left outer join "Production"."ProductReview" pr ON p."ProductID" = pr."ProductID";

```

Product 1 X

select p."Name", coalesce (cast(pr."ProductID" as VARC | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC Name	ABC Рейтинг_продукта
436	LL Road Tire	Нет рейтинга
437	ML Road Tire	Нет рейтинга
438	HL Road Tire	Нет рейтинга
439	Touring Tire	Нет рейтинга
440	LL Mountain Pedal	Нет рейтинга
441	ML Mountain Pedal	Нет рейтинга
442	HL Mountain Pedal	937
443	HL Mountain Pedal	937
444	LL Road Pedal	Нет рейтинга
445	ML Road Pedal	Нет рейтинга
446	HL Road Pedal	Нет рейтинга
447	Touring Pedal	Нет рейтинга

Рисунок 2

На рисунке 3 реализация задачи 3:

--3 Показать список ID людей (поле BusinessEntityID), в котором указано, является человек вендором или нет, из таблиц Purchasing.Vendor, Person.BusinessEntity, используя RIGHT OUTER JOIN.

```

SELECT p."BusinessEntityID"
FROM "Person"."BusinessEntity" p
RIGHT OUTER JOIN "Purchasing"."Vendor" v ON p."BusinessEntityID" = v."BusinessEntityID";

```

BusinessEntity 1 X

SELECT p."BusinessEntityID" FROM "Pers | Введите SQL выражение чтобы отфильтровать результаты

	123 BusinessEntityID
1	1 492
2	1 494
3	1 496
4	1 498
5	1 500
6	1 502
7	1 504
8	1 506
9	1 508
10	1 510
11	1 512
12	1 514
13	1 516

Значение X
1492

Рисунок 3

На рисунке 4 реализация задачи 4:

```
--4 Показать список продуктов (ProductID), которые имеют несколько дат выполнения заказа (DueDate)
-- из Purchasing.PurchaseOrderDetail, используя SELF JOIN.
select pd1."ProductID", pd1."DueDate"
from "Purchasing"."PurchaseOrderDetail" pd1
join "Purchasing"."PurchaseOrderDetail" pd2 on pd1."ProductID" = pd2."ProductID"
--проверяем чтобы значения столбца DueDate не были одинаковыми
and pd1."DueDate" <> pd2."DueDate"
order by pd1."ProductID";
```

PurchaseOrderDetail 1 X

select pd1."ProductID", pd1."DueDate" fr Введите SQL выражение чтобы отфильтровать результаты

	123 ProductID	DueDate
1	1	2014-05-16 00:00:00.000
2	1	2014-05-16 00:00:00.000
3	1	2014-05-16 00:00:00.000
4	1	2014-05-16 00:00:00.000
5	1	2014-05-16 00:00:00.000
6	1	2014-05-16 00:00:00.000
7	1	2014-05-16 00:00:00.000
8	1	2014-05-16 00:00:00.000
9	1	2014-05-16 00:00:00.000
10	1	2014-05-16 00:00:00.000
11	1	2014-05-16 00:00:00.000
12	1	2014-05-16 00:00:00.000
13	1	2014-05-16 00:00:00.000
14	1	2014-05-16 00:00:00.000
15	1	2014-05-16 00:00:00.000
16	1	2014-05-16 00:00:00.000
17	1	2014-05-16 00:00:00.000
18	1	2014-05-16 00:00:00.000

Значение
Edit value:
1
Словарь (Proc
Type part of c
Значение
1
2
3
4
316
317
318
319
320
321
322

Рисунок 4

На рисунке 5 реализация задачи 5:

```
--5 Показать список дат выполнения заказа (DueDate), к которым относятся несколько продуктов (ProductID),
-- из таблицы Purchasing.PurchaseOrderDetail, используя SELF JOIN.
select distinct pd1."DueDate"
from "Purchasing"."PurchaseOrderDetail" pd1
join "Purchasing"."PurchaseOrderDetail" pd2 on pd1."DueDate" = pd2."DueDate" and pd1."ProductID" <> pd2."ProductID";
```

PurchaseOrderDetail 1 X

select distinct pd1."DueDate" from "Purc Введите SQL выражение чтобы отфильтровать результаты

Таблица	DueDate
1	2014-05-02 00:00:00.000
2	2014-07-17 00:00:00.000
3	2014-07-14 00:00:00.000
4	2013-06-12 00:00:00.000
5	2013-09-03 00:00:00.000
6	2014-07-01 00:00:00.000
7	2012-09-11 00:00:00.000
8	2014-10-22 00:00:00.000
9	2014-07-24 00:00:00.000
10	2014-05-19 00:00:00.000
11	2014-01-02 00:00:00.000
12	2014-05-05 00:00:00.000

Значение X
2014-05-02 00:00:00.000

Рисунок 5

На рисунке 6 реализация задачи 6:

--6 Показать товары из таблицы Production.Product, имеющие такой же вес
--как у товара с названием «LL Road Frame - Black, 62», используя self JOIN.

```
select p1.*
from "Production"."Product" p1
join "Production"."Product" p2 on p1."Weight" = p2."Weight"
where p2."Name" = 'LL Road Frame - Black, 62';
```

Product 1

select p1.* from "Production"."Product"

ProductID	Name	ProductNumber	Color	SafetyStockLevel	ReorderPoint	StandardCost
724	LL Road Frame - Black, 62	FR-R38B-62	Black	500	375	204,63
730	LL Road Frame - Red, 62	FR-R38R-62	Red	500	375	187,16

Рисунок 6

На рисунке 7 реализация задачи 7:

--7 Показать комбинированный список таблиц Person.EmailAddress, Person.BusinessEntityAddress
-- по полям BusinessEntityID, rowguid, ModifiedDate, используя UNION.

```
select "BusinessEntityID"
, "rowguid"
, "ModifiedDate"
from "Person"."EmailAddress"
union
select "BusinessEntityID"
, "rowguid"
, "ModifiedDate"
from "Person"."BusinessEntityAddress";
```

Результат 1

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

BusinessEntityID	rowguid	ModifiedDate
6 866	37388f04-4b29-405f-a2e7-fb6b95a6fb3a	2012-11-17 00:00:00.000
14 034	b40a77ab-5483-43b2-ad69-db7bb497b261	2013-04-22 00:00:00.000
12 692	f15d5918-671c-435c-a035-29121baf5301	2012-11-26 00:00:00.000
11 616	aff5e38f-0205-435c-bc70-93ed62a83f85	2014-04-14 00:00:00.000
19 532	101ecda8-e15c-4035-8166-49fc0f8df45e	2013-10-20 00:00:00.000
5 258	37840ef7-d880-4e21-841b-fb9323702233	2013-06-17 00:00:00.000
3 570	d2a405bb-b9b4-4dac-8a0f-6c701f202c3c	2013-09-24 00:00:00.000
19 627	66267c1e-0329-4a30-aed1-1d2daff02fa1	2013-12-13 00:00:00.000
18 654	17435341-3f29-458f-9c8f-9eb77ef743bd	2014-03-18 00:00:00.000
20 064	fdd223ab-40ab-4bcf-96a9-ed2f3a0b9a79	2014-03-10 00:00:00.000
4 455	41f16339-8650-432b-a01f-f18aaf8ea8aa	2014-01-08 00:00:00.000
12 986	9c38ebd4-cb42-43ca-b197-59907e851f4f	2013-10-15 00:00:00.000

Рисунок 7

На рисунке 8 реализация задачи 8:

The screenshot shows a SQL query editor with the following text:

```
--8 Показать список ProductID, которые содержатся и в таблице Production.Product,  
-- и в таблице Production.ProductInventory.  
select "ProductID"  
from "Production"."Product" p  
intersect  
select "ProductID"  
from "Production"."ProductInventory";
```

Below the query editor, there is a results window titled "Результат 1" (Result 1). It shows a table with two columns: "ProductID" and a list of values. The first row is highlighted in blue.

	ProductID
1	933
2	880
3	423
4	442
5	479
6	421
7	411
8	869
9	392
10	429
11	966
12	864
13	760

Рисунок 8

4. ВЫВОД

Изучены операторы, необходимые для работы с данными, находящимися в разных таблицах. Научился создавать запросы на соединение данных из множества таблиц.