ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

«Соединение данных из множества таблиц»

По дисциплине «Информационное обеспечение систем управления»

Выполнил: ст. гр. ТУУ – 411

Каплеева А.В.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Цель работы: 2](#_Toc88061339)

[Формулировка задания: 2](#_Toc88061340)

[Вывод 3](#_Toc88061341)

## Цель работы:

Изучить операторы MS SQL Server, необходимые для работы с данными, находящимися в разных таблицах. Научиться создавать запросы на соединение данных из множества таблиц. Работа производится с учебной базой данной AdwentureWorks2019.

# Формулировка задания:

ВАРИАНТ 9

1. Показать дни рождения (DateBirth) и поля PersonType, FirstName, LastName из таблиц Person.Person, Person.Employee.

-- Показать дни рождения (DateBirth) и поля PersonType, FirstName, LastName из таблиц Person.Person, Person.Employee.  
SELECT p.PersonType  
 , p.FirstName  
 , p.LastName  
 , h.BirthDate  
FROM Person.Person AS p  
INNER JOIN HumanResources.Employee AS h  
ON p.BusinessEntityID = h.BusinessEntityID;

Изображение выглядит как стол

Автоматически созданное описание

Рисунок  1 ‑ Результат выполнения скрипта

1. Показать список продуктов (поле Name), в котором указано, есть ли у продукта количество заказов или нет, из таблиц Production.Product, Sales. SalesOrderDetail, используя LEFT OUTER JOIN.

-- Показать список продуктов (поле Name), в котором указано, есть ли у продукта количество заказов или нет, из таблиц Production.Product, Sales. SalesOrderDetail, используя LEFT OUTER JOIN.  
SELECT p.[Name]  
 , s.OrderQty  
FROM Production.Product AS p  
LEFT OUTER JOIN Sales.SalesOrderDetail AS s  
ON p.ProductID = s.ProductID  
WHERE s.OrderQty IS NOT NULL;

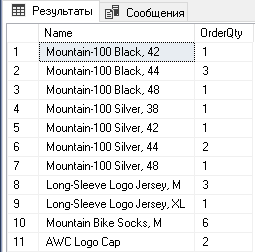


Рисунок  2 ‑ Результат выполнения скрипта

1. Показать список товаров (поле Name), в котором указан, есть ли у товара полка (поле Shelf) или нет, из таблицы Production.ProductInventory, Production.Product, используя RIGHT OUTER JOIN.

-- Показать список товаров (поле Name), в котором указан, есть ли у товара полка (поле Shelf) или нет, из таблицы Production.ProductInventory, Production.Product, используя RIGHT OUTER JOIN.  
SELECT p.[Name]  
 , i.Shelf  
FROM Production.ProductInventory AS i  
RIGHT OUTER JOIN Production.Product AS p  
ON i.ProductID = p.ProductID;

Изображение выглядит как текст, квитанция

Автоматически созданное описание

Рисунок  3 – Результат выполнения скрипта

1. Показать список покупателей (CustomerID), которые имеют несколько дат оформления заказа (OrderDate), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.

-- Показать список покупателей (CustomerID), которые имеют несколько дат оформления заказа (OrderDate), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.  
SELECT DISTINCT s1.CustomerID  
 , s1.OrderDate  
FROM Sales.SalesOrderHeader AS s1  
INNER JOIN Sales.SalesOrderHeader AS s2  
ON s1.CustomerID = s2.CustomerID  
AND s1.OrderDate <> s2.OrderDate;

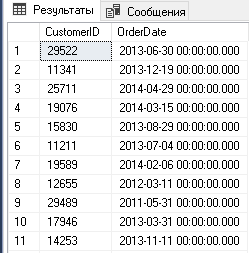


Рисунок  4 ‑ Результат выполнения скрипта

1. Показать список дат оформления заказа (OrderDate), которые относятся к нескольким покупателям (CustomerID), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.

-- Показать список дат оформления заказа (OrderDate), которые относятся к нескольким покупателям (CustomerID), из таблицы Sales.SalesOrderHeader, используя SELF JOIN.  
SELECT DISTINCT s1.OrderDate  
 , s1.CustomerID  
FROM Sales.SalesOrderHeader AS s1  
INNER JOIN Sales.SalesOrderHeader AS s2  
ON s1.OrderDate = s2.OrderDate  
AND s1.CustomerID <> s2.CustomerID;

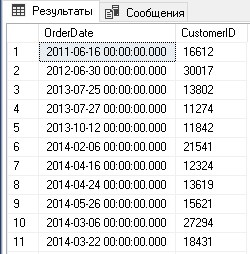


Рисунок  5 – Результат выполнения скрипта

1. Показать список продавцов (SalesPersonID из Sales.SalesOrderHeader), которые имеют нескольких покупателей, CustomerID которых начинается с 3, используя SELF JOIN.

-- Показать список продавцов (SalesPersonID из Sales.SalesOrderHeader), которые имеют нескольких покупателей, CustomerID которых начинается с 3, используя SELF JOIN.  
SELECT DISTINCT s1.SalesPersonID  
 , s1.CustomerID  
FROM Sales.SalesOrderHeader AS s1  
INNER JOIN Sales.SalesOrderHeader AS s2  
ON s1.SalesPersonID = s2.SalesPersonID  
AND s1.CustomerID <> s2.CustomerID  
WHERE s1.CustomerID >= 3  
AND s2.CustomerID >= 3;

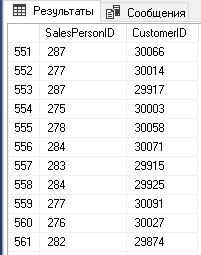


Рисунок  6 ‑ Результат выполнения скрипта

1. Показать комбинированный список таблиц Production.Product, Production.ProductSubcategory по полям Name, ProductSubcategoryID, используя UNION.

-- Показать комбинированный список таблиц Production.Product, Production.ProductSubcategory по полям Name, ProductSubcategoryID, используя UNION.  
SELECT [Name]  
 , ProductSubcategoryID  
FROM Production.Product  
UNION  
SELECT [Name]  
 , ProductSubcategoryID  
FROM Production.ProductSubcategory;

Изображение выглядит как стол

Автоматически созданное описание

Рисунок  7 – Результат выполнения скрипта

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

-- Показать список ProductID, которые содержатся и в таблице Production.Product, и в таблице Production.ProductDocument.  
SELECT ProductID  
FROM Production.Product  
INTERSECT   
SELECT ProductID  
FROM Production.ProductDocument;

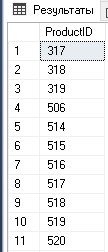


Рисунок  8 ‑ Результат выполнения скрипта

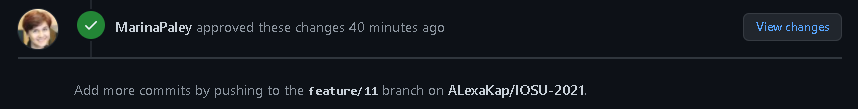


Рисунок  9 – Результат проверки скрипта

## Вывод:

Изучены операторы MS SQL Server, необходимые для работы с данными, находящимися в разных таблицах. Получен навык создания запросов на соединение данных из множества таблиц. Работа производилась с учебной базой данной AdwentureWorks2019.