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 №2

по дисциплине

«Информационное обеспечение систем управления»

ВАРИАНТ 2

Выполнил: ст. гр. ТУУ-411

Гусева Ольга Олеговна

Проверил: к.т.н., доц. Васильева М. А.

Москва 2022

**Цель работы:** Изучить операторы MS SQL SERVER, необходимые для группировки и обобщения данных. Научиться создавать запросы с агрегатными функциями. Работа производится с учебной базой данной AdwentureWorks.

## Формулировка задания

1. Найти минимальную цену товара (ListPrice) из таблицы Production.Product.
2. Найти средний вес (Weight) из таблицы Production.Product, не учитывать пустые значения.
3. Для каждого пола найти суммарное количество часов отпуска из таблицы (поставить фильтр: работник был принят до 2008 года) HumanResources.Employee, не учитывать пустые значения.
4. Вывести пол, где среднее количество часов выходных больше 50 из HumanResources.Employee.
5. Вывести BusinessEntityID, LastReceiptDate и сумму по StandardPrice для BusinessEntityID с разными LastReceiptDate, предусмотреть вывод общей суммы для всех StandardPrice у различных BusinessEntityID из таблицы Purchasing.ProductVendor. (Использовать ROOLUP)

## Основная часть

1. Скрипт для Задания №1 и результат его выполнения (Рисунок 1).

--1. Найти минимальную цену товара (ListPrice) из таблицы

--Production.Product

SELECT MIN(ListPrice) AS MinPrice

FROM Production.Product

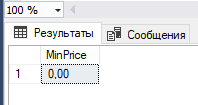


Рисунок 1 - Результат выполнения скрипта для Задания №1

1. Скрипт для Задания №2 и результат его выполнения (Рисунок 2).

--2. Найти средний вес (Weight) из таблицы Production.Product, не

--учитывать пустые значения.

SELECT AVG([Weight]) AS AvgWeight

FROM Production.Product

WHERE [Weight] IS NOT NULL;

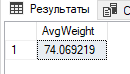


Рисунок 2 - Результат выполнения скрипта для Задания №2

1. Скрипт для Задания №3 и результат его выполнения (Рисунок 3).

--3. Для каждого пола найти суммарное количество часов отпуска из

--таблицы (поставить фильтр: работник был принят до 2008 года)

--HumanResources.Employee, не учитывать пустые значения.

SELECT SUM(VacationHours) AS SumVacationHours

, Gender

FROM HumanResources.Employee

WHERE HireDate <'2008' AND

VacationHours IS NOT NULL

GROUP BY Gender

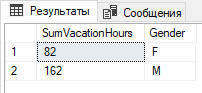


Рисунок 3 - Результат выполнения скрипта для Задания №3

1. Скрипт для Задания №4 и результат его выполнения (Рисунок 4).

--4. Вывести пол, где среднее количество часов выходных больше 50 из

--HumanResources.Employee.

SELECT Gender

, AVG(VacationHours) AS AvgVacantionHours

FROM HumanResources.Employee

GROUP BY Gender

HAVING AVG(VacationHours)>50;

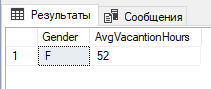


Рисунок 4 - Результат выполнения скрипта для Задания №4

1. Скрипт для Задания №5 и результат его выполнения (Рисунок 5).

--5. Вывести BusinessEntityID, LastReceiptDate и сумму по

--StandardPrice для BusinessEntityID с разными LastReceiptDate,

--предусмотреть вывод общей суммы для всех StandardPrice у различных

--BusinessEntityID из таблицы Purchasing.ProductVendor. (Использовать --ROOLUP)

SELECT BusinessEntityID

, LastReceiptDate

, SUM(StandardPrice) AS SumStandartPrice

FROM Purchasing.ProductVendor

GROUP BY

ROLLUP (BusinessEntityID, LastReceiptDate);

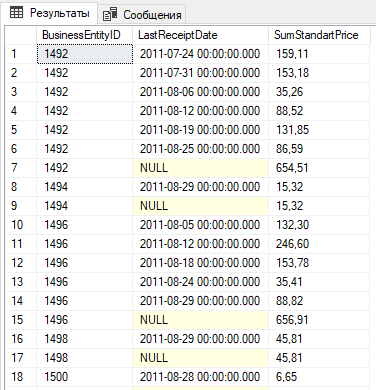
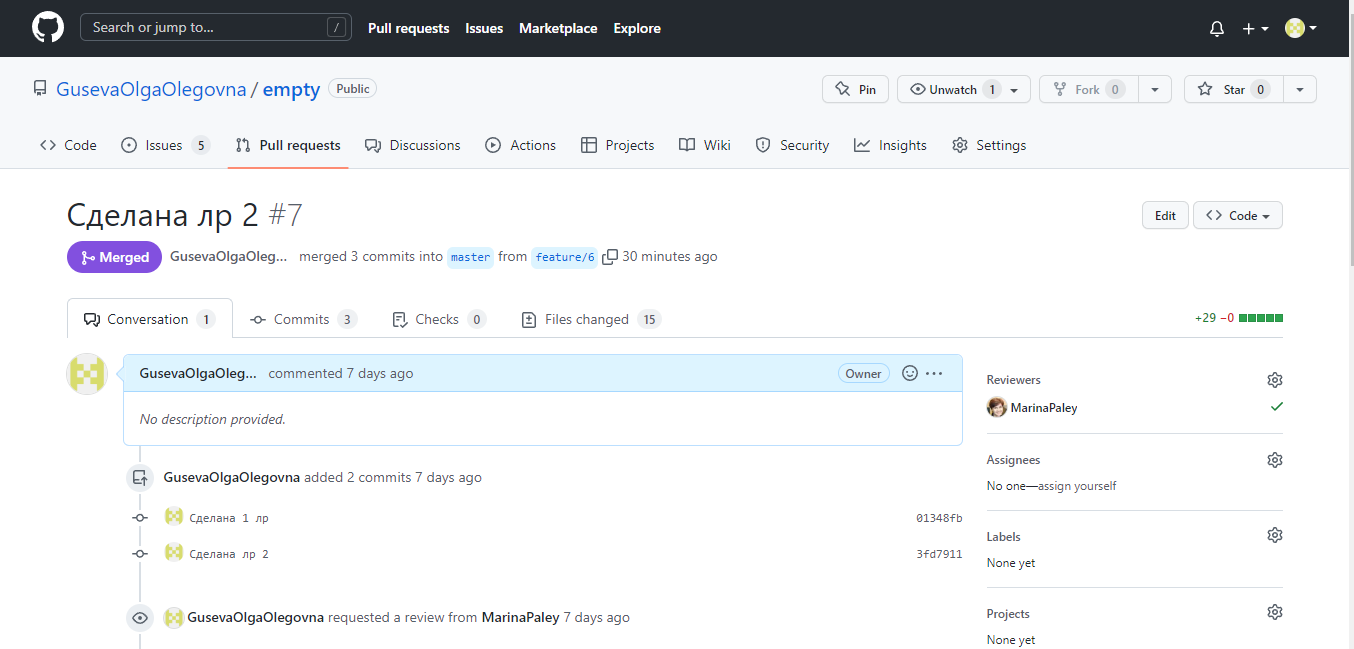


Рисунок 5 - Результат выполнения скрипта для Задания №5

## Оценка преподавателя



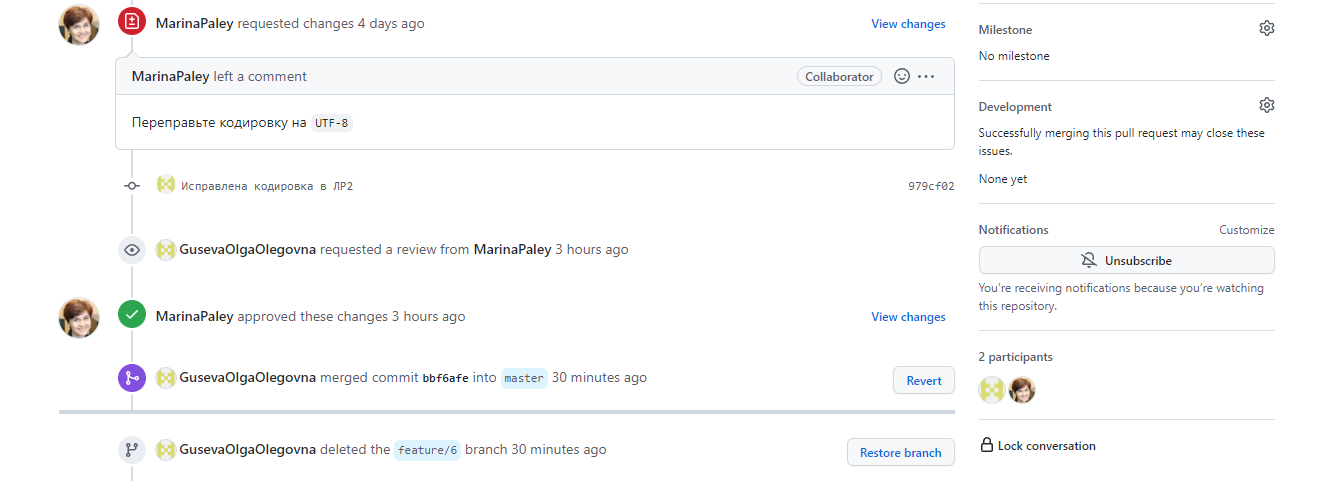


Рисунок 6 - Снимок экрана, на котором виден approve и автор репозитория, которому данный approve поставлен