МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

дисциплина «Базы данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

«Создание запросов к реляционной базе данных с использование графических средств»

Вариант №5

Выполнил: студент гр. ИТИ-22

Карпенко Д.Е.

Принял: преподаватель-стажёр

Карась О.В.

Гомель 2023

**Цель работы:** изучить основы работы с запросами к реляционной базе данных с использованием графических средств SQL Server Management Studio.

**Задание:**

**1. Создание простых запросов на выборку.**

**1.1.** Создать простой многотабличный запрос, позволяющий выводить данные из таблицы «Payments» с одновременным выводом данных о студенте. Запрос должен содержать поля StudentId, Surname, Name, MiddleName, взятые из таблицы «Students», и поля PaymentDate, Amount, PurposeId, взятые из таблицы «Payments».

**1.2.** Создать простой многотабличный запрос для вывода списка улиц города Гомеля. Запрос должен содержать поля: Name (таблица «Towns») и Name (таблица «Streets»). Связь между таблицами установить по полю TownId. В бланке запроса для поля Name (таблица «Towns») в столбец Filter внести - Гомель.

**2. Группировка данных в запросах на выборку.**

**2.1.** Создать многотабличный запрос (представление) с именем «ViewPaymentSum1» для подсчета общих сумм оплат студентов за весь период оплат. Запрос должен содержать поля Фамилия, Имя, Отчество, взятые из таблицы «Students», и поле Amount, взятое из таблицы «Payments». При создании запроса использовать команду Group By. В качестве групповой операции для поля Amount в раскрывающемся списке указать Sum (суммирование), а для остальных полей – Group By.

**2.2.** Создать многотабличный запрос (представление) с именем «ViewPaymentSum2» путем копирования и последующего редактирования представления «ViewPaymentSum1». Выполнить следующие изменения: 1) добавить в бланк запроса поле Group; 2) удалить из бланка запроса поля Surname, Name, MiddleName; 3) выполнить для поля Group установку: Group By – Group By.

**2.3.** Создать многотабличный запрос (представление) с именем «ViewPaymentSum3» путем редактирования запроса «ViewPaymentSum2». Выполнить следующие изменения установок для поля Group: Group By – Where; Выход – нет (снять флажок); Filter: указать номер группы из имеющихся в БД.

Сравнить результаты представлений (Views) «ViewPaymentSum1», «ViewPaymentSum2», «ViewPaymentSum3».

**3. Создание многотабличных запросов на выборку с условиями и запросов на изменение данных**

**3.1. Запрос на выборку из нескольких таблиц с условием**

Создать следующие запросы с условиями:

1. отбор данных (личные данные, платежи) о студентах, живущих в некотором городе;
2. вывод данных (личные данные, платежи) о некотором студенте;
3. вывод данных о студентах из заданного города, поступивших на обучение в некотором диапазоне лет (от некоторого начального года до некоторого конечного года).

**3.2. Выполнение вычислений в запросе на выборку**

Создать запрос на выборку, содержащий следующие базовые и вычисляемые поля: Surname, Name, MiddleName, Фактическая сумма оплаты: Sum([Amount]) (вычисляемое поле), Процент оплаты: Sum([Amount])\*100/100000 (вычисляемое поле).

**3.3. Запросы на изменение данных**

Создать запрос на удаление: удалить из таблицы «Payments» все данные об оплатах с датой до 01.01.2019 (или подобрать самостоятельно).

Создать запрос на обновление: обновить в таблице «Students» название некоторого факультета (использовать некоторое значение).

Протестировать правильность выполнения запросов.

**4. Задания, выполняемые согласно номеру своего варианта.**

**4.1. Создание простых многотабличных запросов**

Создать многотабличный запрос для вывода следующих сведений о студентах (для факультета выводить полное название): фамилия, имя, отчество, дата рождения, факультет;

**4.2. Создание запросов на выборку с группировкой результатов**

Создать запрос для нахождения следующих итоговых показателей, вычисляемых по группам записей (используются группировка данных в запросах и итоговые функции Sum, Avg, Min, Max, Count, First, Last): минимальное значение суммы оплаты при одном платеже для каждого студента

**4.3. Создание многотабличных запросов на выборку с условиями и вычисляемыми полями**

Создать и протестировать запросы, содержащие следующие сведения (при необходимости, использовать итоговые запросы и запрос с вычисляемыми полями, содержащий поля итоговых запросов): разность между суммами всех платежей студента 1 и студента 2;

**4.4. Создание запросов на изменение данных**

Создать и протестировать запросы, выполняющие: обновление в таблице «Улицы» названия некоторой улицы (использовать параметры);

**Ход работы**

Запросы написаны на языке QBE (построение по образцу), где:

1) USE [] – выбирается база данных;

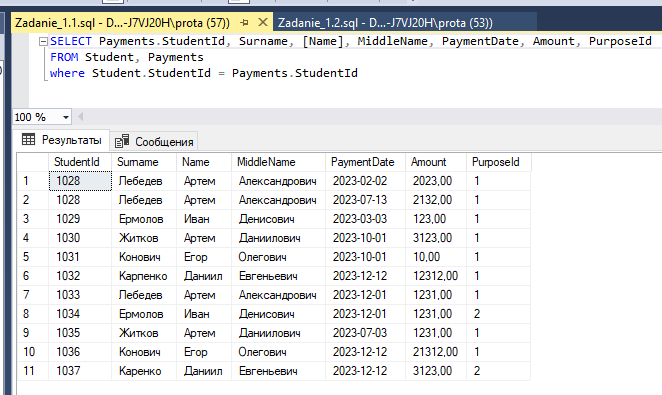
2) SELECT – возвращающий набор данных (выборку) из базы данных;

3) FROM – таблица, которая является поставщиком данных;

4) WHERE – описание условия запроса;

5) Group By – группировка объектов.

На рисунке 1 представлен запрос, удовлетворяющий заданию 1.1. Запрос берет необходимые столбцы из двух таблиц и выводит их соответственно индексам.

****

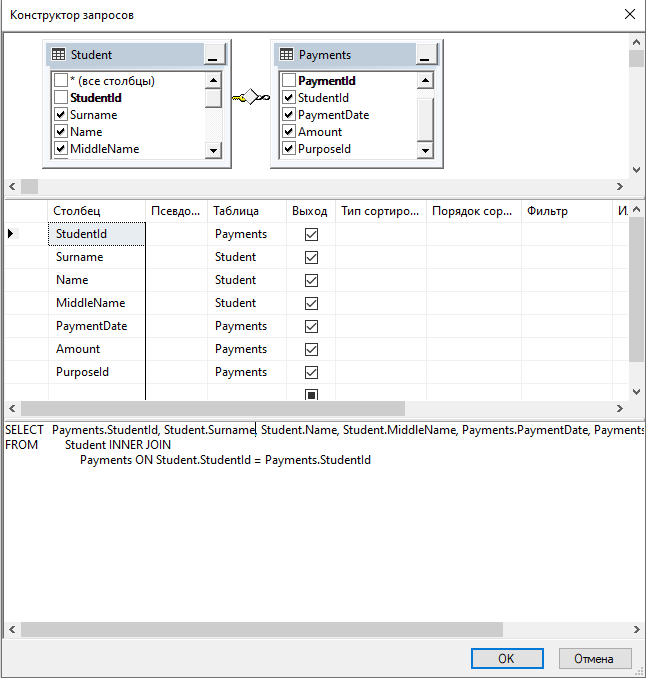
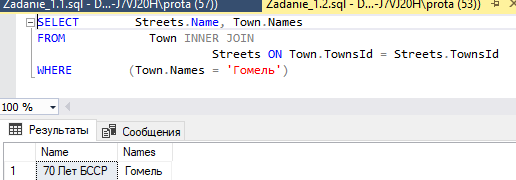
****

Рисунок 1 – Выполнение запроса задания 1.1

На рисунке 2 представлен запрос, удовлетворяющий заданию 1.2. Запрос выбирает соответствующие условию данных и выводит название улицы, расположенные в городе.

****

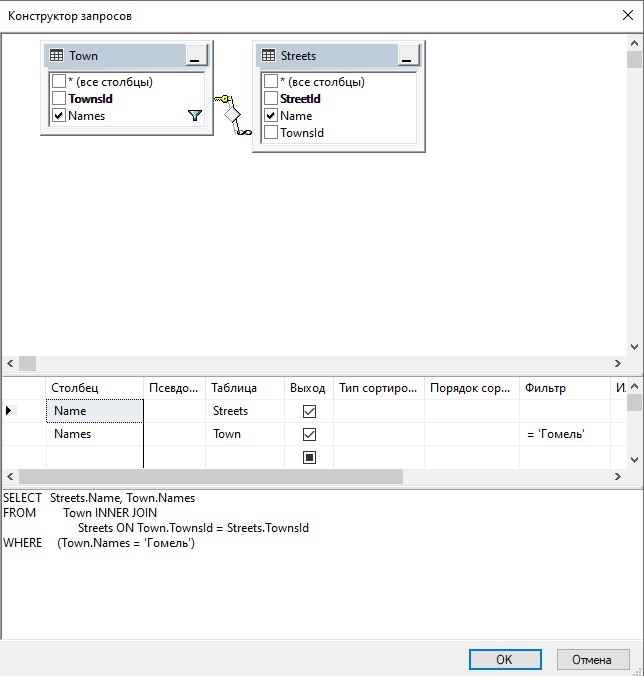
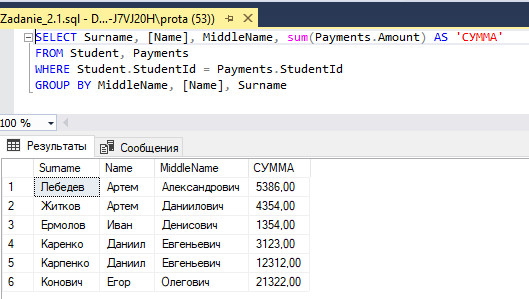
****

Рисунок 2 – Выполнение запроса задания 1.2

На рисунке 3 представлен запрос, удовлетворяющий заданию 2.1Запрос выводит сумму платежей всех студентов.



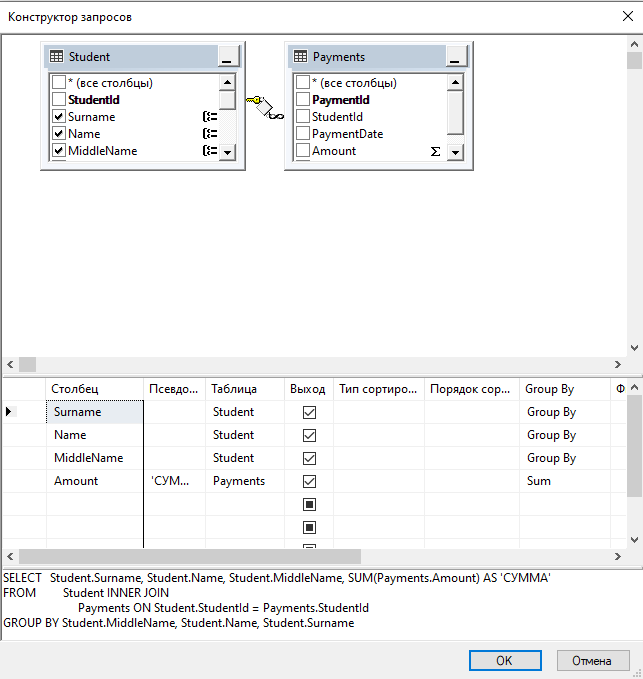
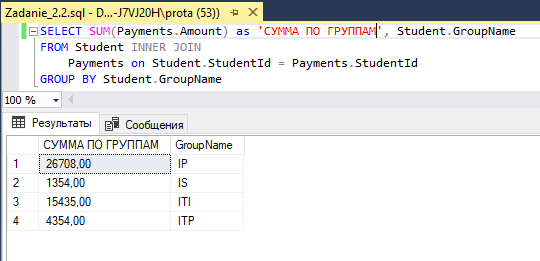


Рисунок 3 **–** Выполнение запроса задания 2.1

На рисунке 3 представлен запрос, удовлетворяющий заданию 2.2. Запрос является редактированной версией задания 2.1.



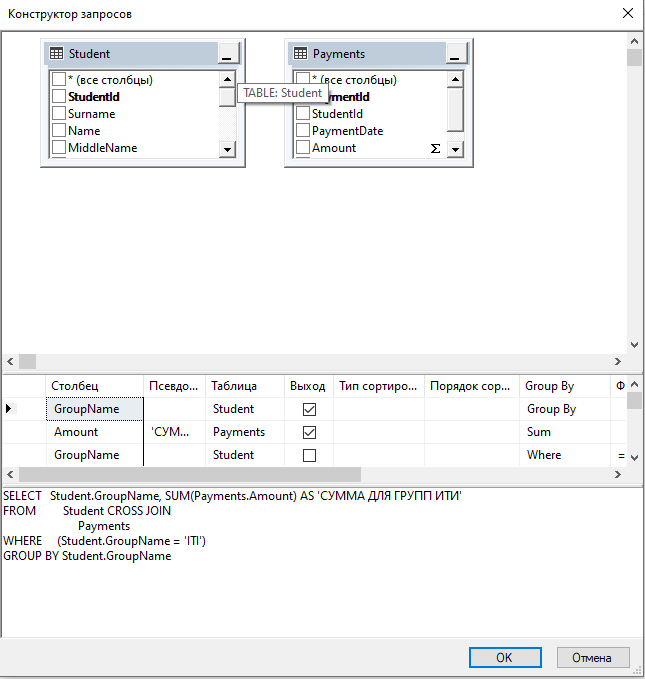
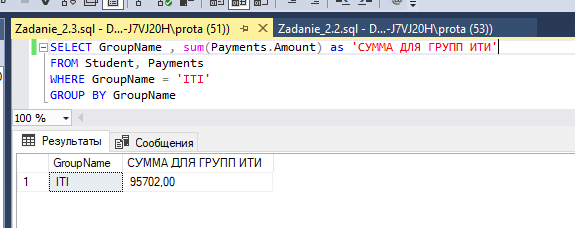


Рисунок 4 **–** Выполнение запроса задания 2.2

На рисунке 3 представлен запрос, удовлетворяющий заданию 2.3. Запрос является редактированной версией задания 2.2.



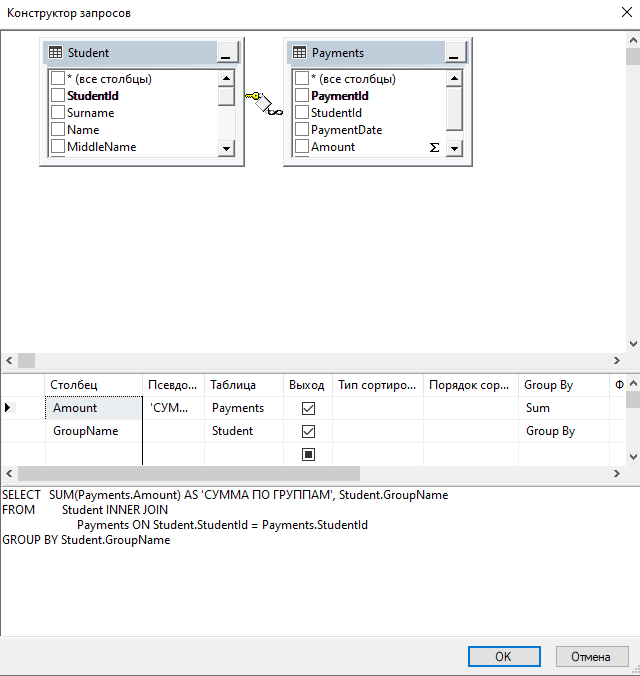
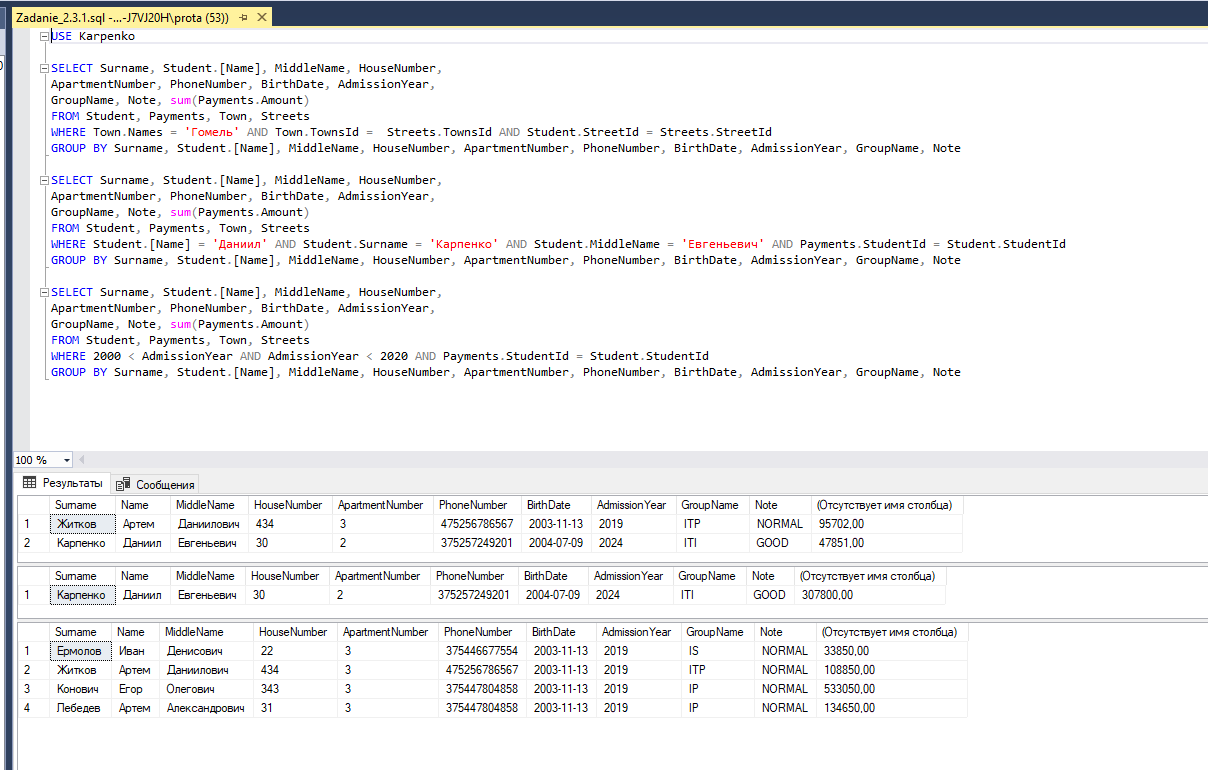
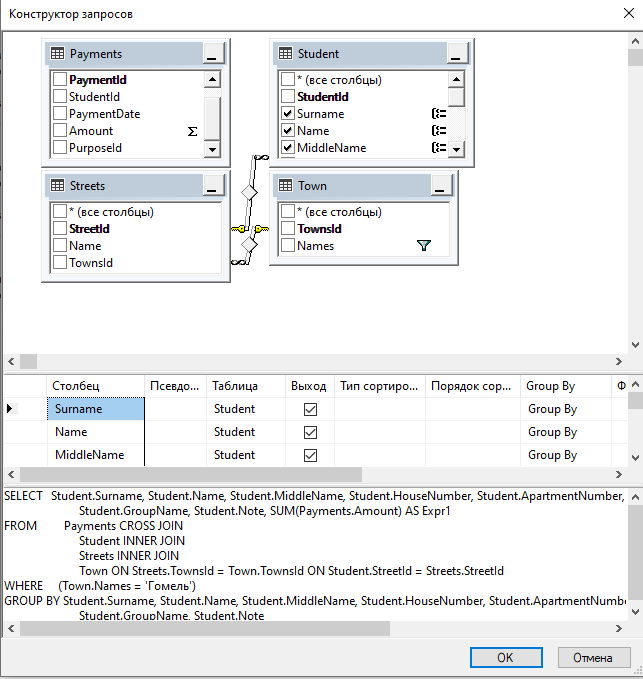
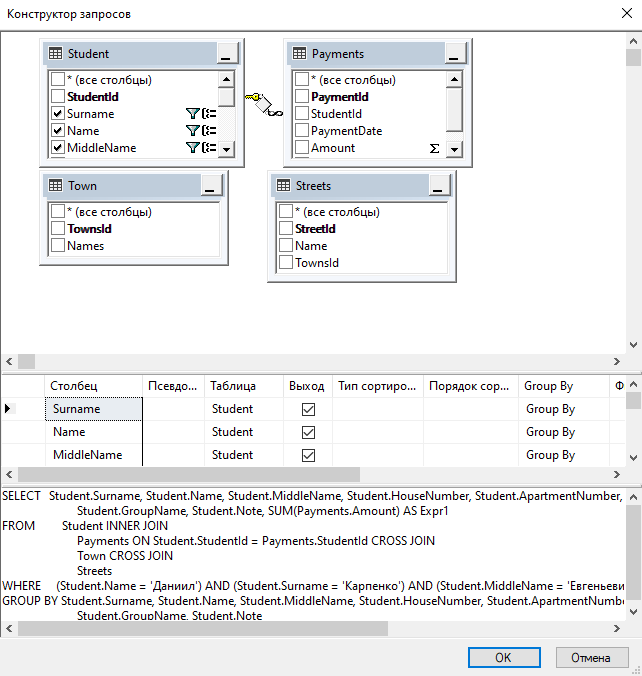


Рисунок 5 - Выполнение запроса задания 2.3

На рисунке 6 представлен запрос, удовлетворяющий заданию 3.1. Запрос высчитывает сумму платежей выбранных студентов и выводит общие данные.

****

****

****

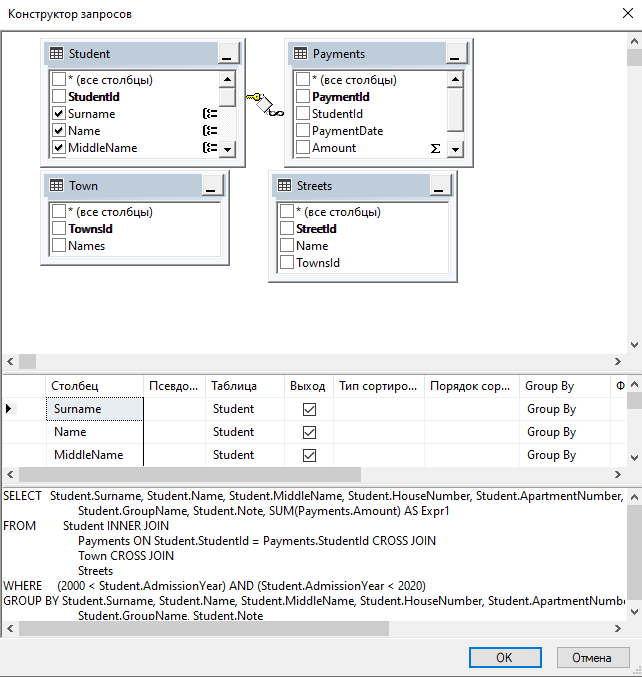
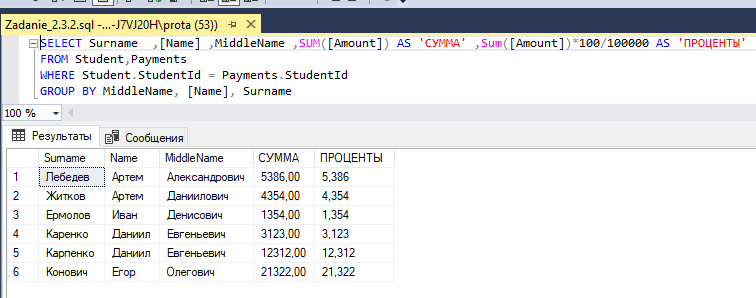
****

Рисунок 6 – Выполнение запроса задания 3.1

На рисунке 7 представлен запрос, удовлетворяющий заданию 3.2. Запрос высчитывает сумму платежей выбранных студентов и процент, высчитанный по формуле.

****

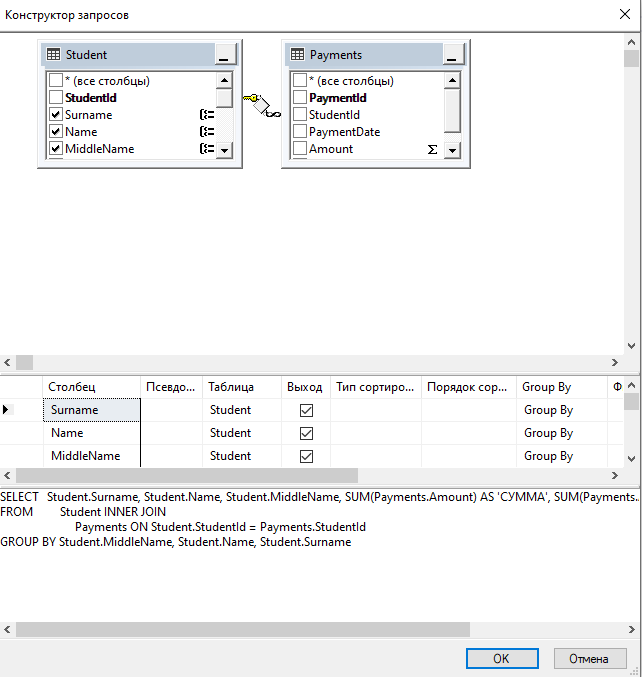
****

Рисунок 7 – Выполнение запроса задания 3.2

На рисунке 8 представлен запрос, удовлетворяющий заданию 3.3. Запрос удаляет поле и обновляет поле, удовлетворяющее условию.

1. DELETE – удаление столбца из таблицы;
2. UPDATE – обновление столбца таблицы.

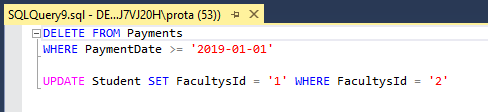
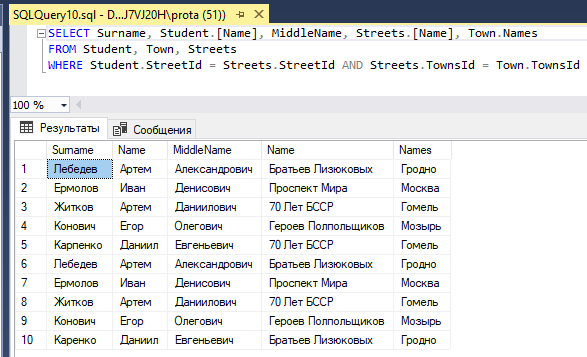
****

Рисунок 8 **–** Выполнение запроса задания 3.3

На рисунке 9 представлен запрос, удовлетворяющий заданию 4.1. Запрос выводит данные о студенте, ФИО, название города и улицы.

****

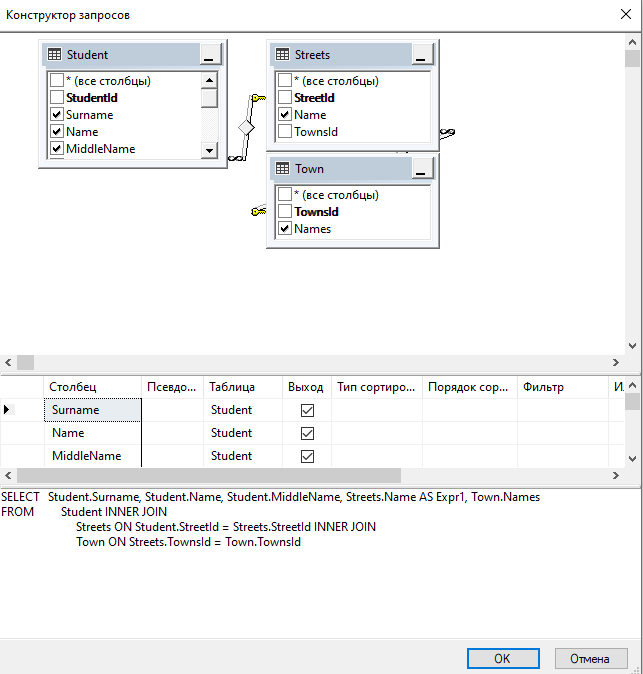
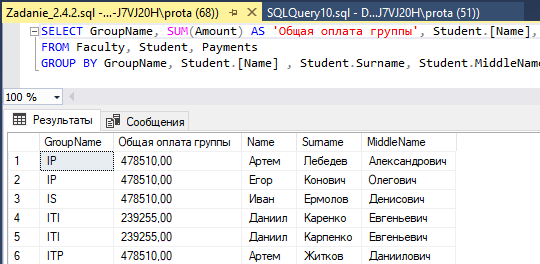


Рисунок 9 **–** Выполнение запроса задания 4.1

На рисунке 10 представлен запрос, удовлетворяющий заданию 4.2. Запрос выводит Ф.И.О. студента, сумму всех оплат и группу.

****

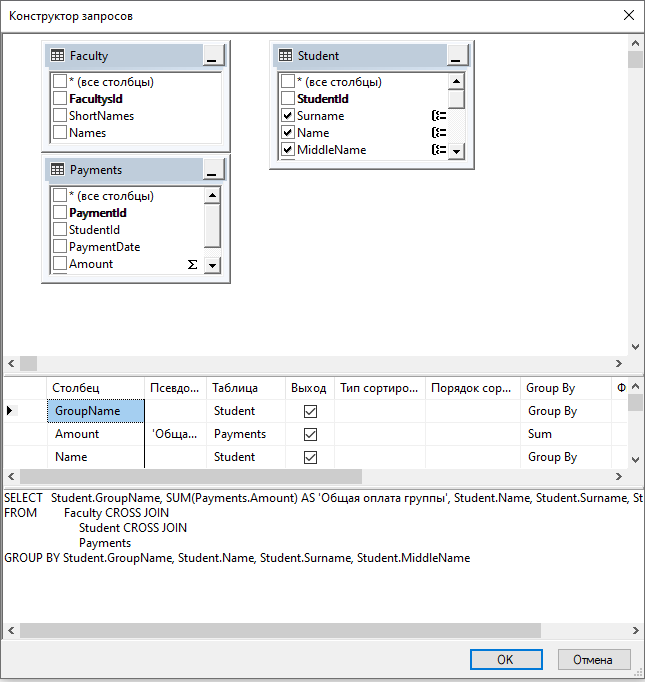
****

Рисунок 10 **–** Выполнение запроса задания 4.2

На рисунке 11 представлен запрос, удовлетворяющий заданию 4.3, также результат выполнения и конструктор запроса. Запрос выводит разность сумм оплат двух групп.

1. DECLARE – объявление переменных;
2. SET – присваивание значения переменным.

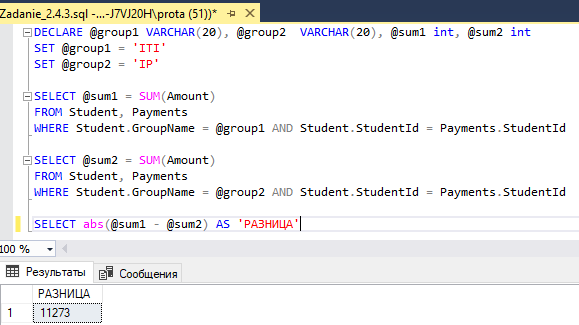


Рисунок 11**–** Выполнение запроса задания 4.3

На рисунке 12 представлен запрос, удовлетворяющий заданию 4.4. Запрос удаляет из таблицы данные от тех студентах которые не провели оплаты в 2023-01-01.

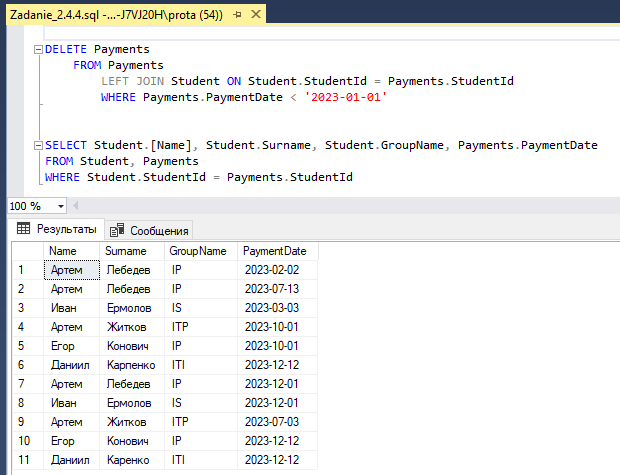


Рисунок 12**–** Выполнение запроса задания 4.4

**Вывод:** в результате выполнения лабораторной работы были получены знания в создании многотабличных запросов QBE и использование условий и группировки данных.

**ПРИЛОЖЕНИЕ А**

**Текст скриптов**

**Файл Zadanie\_1.1.sql:**

SELECT Payments.StudentId, Surname, [Name], MiddleName, PaymentDate, Amount, PurposeId

FROM Student, Payments

where Student.StudentId = Payments.StudentId

**Файл Zadanie\_1.2.sql:**

SELECT Streets.Name, Town.Names

FROM Town INNER JOIN

Streets ON Town.TownsId = Streets.TownsId

WHERE (Town.Names = 'Гомель')

**Файл Zadanie\_2.1.sql:**

SELECT Surname, [Name], MiddleName, sum(Payments.Amount) AS 'СУММА'

FROM Student, Payments

WHERE Student.StudentId = Payments.StudentId

GROUP BY MiddleName, [Name], Surname

**Файл Zadanie\_2.2.sql:**

SELECT SUM(Payments.Amount) as 'СУММА ПО ГРУППАМ', Student.GroupName

FROM Student INNER JOIN

Payments on Student.StudentId = Payments.StudentId

GROUP BY Student.GroupName

**Файл Zadanie\_2.2.3.sql:**

SELECT GroupName , sum(Payments.Amount) as 'СУММА ДЛЯ ГРУПП ИТИ'

FROM Student, Payments

WHERE GroupName = 'ITI'

GROUP BY GroupName

**Файл Zadanie\_2.3.1.sql:**

USE Karpenko

SELECT Surname, Student.[Name], MiddleName, HouseNumber,

ApartmentNumber, PhoneNumber, BirthDate, AdmissionYear,

GroupName, Note, sum(Payments.Amount)

FROM Student, Payments, Town, Streets

WHERE Town.Names = 'Гомель' AND Town.TownsId = Streets.TownsId AND Student.StreetId = Streets.StreetId

GROUP BY Surname, Student.[Name], MiddleName, HouseNumber, ApartmentNumber, PhoneNumber, BirthDate, AdmissionYear, GroupName, Note

SELECT Surname, Student.[Name], MiddleName, HouseNumber,

ApartmentNumber, PhoneNumber, BirthDate, AdmissionYear,

GroupName, Note, sum(Payments.Amount)

FROM Student, Payments, Town, Streets

WHERE Student.[Name] = 'Даниил' AND Student.Surname = 'Карпенко' AND Student.MiddleName = 'Евгеньевич' AND Payments.StudentId = Student.StudentId

GROUP BY Surname, Student.[Name], MiddleName, HouseNumber, ApartmentNumber, PhoneNumber, BirthDate, AdmissionYear, GroupName, Note

SELECT Surname, Student.[Name], MiddleName, HouseNumber,

ApartmentNumber, PhoneNumber, BirthDate, AdmissionYear,

GroupName, Note, sum(Payments.Amount)

FROM Student, Payments, Town, Streets

WHERE 2000 < AdmissionYear AND AdmissionYear < 2020 AND Payments.StudentId = Student.StudentId

GROUP BY Surname, Student.[Name], MiddleName, HouseNumber, ApartmentNumber, PhoneNumber, BirthDate, AdmissionYear, GroupName, Note

**Файл Zadanie\_2.3.2.sql:**

SELECT Surname ,[Name] ,MiddleName ,SUM([Amount]) AS 'СУММА' ,Sum([Amount])\*100/100000 AS 'ПРОЦЕНТЫ'

FROM Student,Payments

WHERE Student.StudentId = Payments.StudentId

GROUP BY MiddleName, [Name], Surname

**Файл Zadanie\_2.3.3.sql:**

DELETE FROM Payments

WHERE PaymentDate >= '2019-01-01'

UPDATE Student SET FacultysId = '1' WHERE FacultysId = '2'

**Файл Zadanie\_2.4.1.sql:**

SELECT Surname, Student.[Name], MiddleName, Streets.[Name], Town.Names

FROM Student, Town, Streets

WHERE Student.StreetId = Streets.StreetId AND Streets.TownsId = Town.TownsId

**Файл Zadanie\_2.4.2.sql:**

SELECT GroupName, SUM(Amount) AS 'Общая оплата группы', Student.[Name], Student.Surname, Student.MiddleName

FROM Faculty, Student, Payments

GROUP BY GroupName, Student.[Name] , Student.Surname, Student.MiddleName

**Файл Zadanie\_2.4.3.sql:**

DECLARE @group1 VARCHAR(20), @group2 VARCHAR(20), @sum1 int, @sum2 int

SET @group1 = 'ITI'

SET @group2 = 'IP'

SELECT @sum1 = SUM(Amount)

FROM Student, Payments

WHERE Student.GroupName = @group1 AND Student.StudentId = Payments.StudentId

SELECT @sum2 = SUM(Amount)

FROM Student, Payments

WHERE Student.GroupName = @group2 AND Student.StudentId = Payments.StudentId

SELECT abs(@sum1 - @sum2) AS 'РАЗНИЦА'

**Файл Zadanie\_2.4.4.sql:**

DELETE Payments

FROM Payments

LEFT JOIN Student ON Student.StudentId = Payments.StudentId

WHERE Payments.PaymentDate < '2023-01-01'

SELECT Student.[Name], Student.Surname, Student.GroupName, Payments.PaymentDate

FROM Student, Payments

WHERE Student.StudentId = Payments.StudentId