НАО «Карагандинский Технический Университет имени Абылкаса Сагинова»

Кафедра «Кибербезопасности и искусственного интеллекта»

Лабораторная работа №3

по дисциплине «Введение в базы данных»

Тема: Создание запросов и модификация таблиц базы данных

Принял:

преп. Жакина М. М.

Выполнил:

студ. гр. СИБ-23-7 Бобер О.Н.

Караганда 2025

**Цель работы:** усвоить способы создания выборки данных в среде СУБД MS SQL Server.

**Содержание отчета:**

1. Название и цель работы
2. Индивидуальное задание
3. Скрипты запросов

**Индивидуальное задание**

1. Реализовать запросы в соответствии с требованиями индивидуального варианта задания на разработку курсового проекта

**Запросы из индивидуального задания:**

1. Перечень групп, студенты которых защищались «*I-го*» числа
2. Количество студентов, защитившие дипломные проекты по каждой из специальностей
3. Список руководителей проектов студентов, защитивших дипломы на отлично
4. Средний балл по результат защиты студентов каждой из групп ВУЗа
5. Категории дипломных проектов, защита которых проходила «*I-го*» числа

**Запросы:**

SELECT DISTINCT g.group\_name

FROM Groups g

JOIN Students s ON g.group\_code = s.group\_code

JOIN Projects\_Defense\_Schedule pds ON s.student\_code = pds.student\_code

WHERE DAY(pds.defense\_date) = 1;

SELECT sp.specialty\_name, COUNT(DISTINCT s.student\_code) AS students\_count

FROM Students s

JOIN Specialties sp ON s.specialty\_code = sp.specialty\_code

JOIN Projects\_Defense\_Schedule pds ON s.student\_code = pds.student\_code

GROUP BY sp.specialty\_name;

SELECT DISTINCT pr.professor\_surname, pr.professor\_first\_name, pr.professor\_patronymic

FROM Professors pr

JOIN Projects p ON pr.professor\_code = p.professor\_code

JOIN Projects\_Defense\_Schedule pds ON p.student\_code = pds.student\_code

JOIN GAK\_Results gr ON pds.defense\_date = gr.defense\_date

WHERE gr.grade = 5;

SELECT g.group\_name, AVG(gr.grade) AS average\_grade

FROM Groups g

JOIN Students s ON g.group\_code = s.group\_code

JOIN Projects\_Defense\_Schedule pds ON s.student\_code = pds.student\_code

JOIN GAK\_Results gr ON pds.defense\_date = gr.defense\_date

GROUP BY g.group\_name;

SELECT DISTINCT pct.project\_category\_type\_name

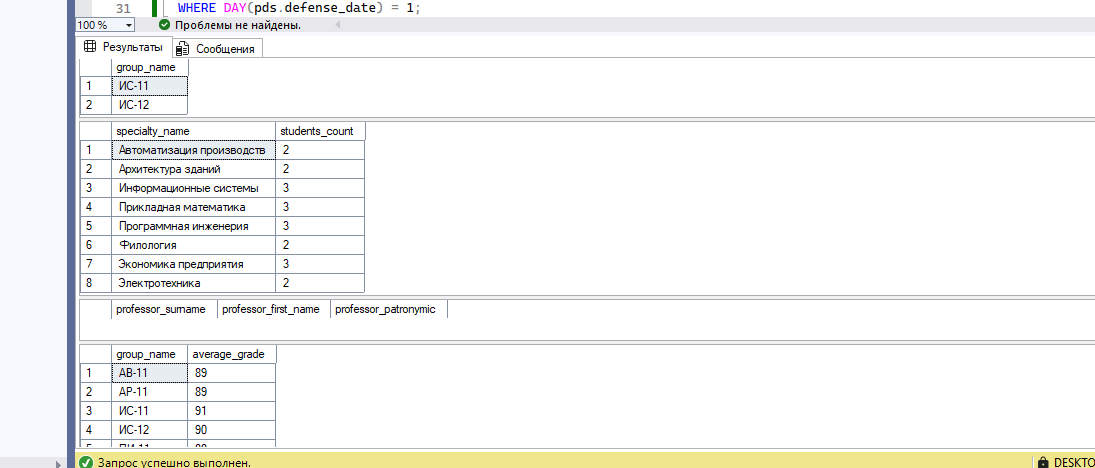
FROM Projects\_Categories\_Types pct

JOIN Projects p ON pct.project\_category\_type\_code = p.project\_category\_type\_code

JOIN Projects\_Defense\_Schedule pds ON p.student\_code = pds.student\_code

WHERE DAY(pds.defense\_date) = 1;

**Результат:**

****

**Контрольные вопросы**

1. Возможности меню Query Editor

Query Editor — это встроенный инструмент SQL Server Management Studio, предназначенный для написания, редактирования и выполнения SQL-запросов.  
Он предоставляет разработчику полный контроль над взаимодействием с базой данных.

Основные возможности:

1. Создание и редактирование SQL-запросов  
   Позволяет писать инструкции SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP и др.
2. Выполнение запросов
   * Запуск выделенного текста (Execute Selected Query)
   * Выполнение всего скрипта (Execute или клавиша F5)
3. Отображение результатов
   * В виде таблицы (Grid)
   * В виде текста (Text)
   * В виде файлов (File output)
4. Подсветка синтаксиса и автоформатирование  
   Облегчает чтение и отладку кода.
5. Проверка корректности синтаксиса  
   Команда Parse (или Ctrl + F5) проверяет ошибки в запросе без выполнения.
6. Использование переменных и параметров  
   Можно объявлять переменные с помощью DECLARE и присваивать значения через SET.
7. План выполнения (Execution Plan)  
   Отображает, как SQL Server оптимизирует выполнение запроса — используется для анализа производительности.
8. Интеграция с транзакциями  
   Можно использовать операторы BEGIN TRAN, COMMIT, ROLLBACK.
9. Отладка и профилирование  
   Позволяет выполнять запросы пошагово и отслеживать их влияние на данные.
10. Подключение к разным базам данных  
    Из одного окна можно переключаться между базами (USE <database\_name>).

2. Конструкция оператора SELECT

Оператор SELECT — основной инструмент для извлечения данных из таблиц базы данных.

Общая структура:

SELECT [DISTINCT]

список\_столбцов

FROM

таблица\_или\_представление

[JOIN тип\_соединения другая\_таблица ON условие]

[WHERE условие\_отбора]

[GROUP BY список\_полей]

[HAVING условие\_для\_групп]

[ORDER BY поле [ASC|DESC]];

Компоненты:

1. SELECT — указывает, какие столбцы выводить.  
   Пример:
2. SELECT student\_surname, gpa FROM Students;
3. DISTINCT — убирает повторяющиеся строки.
4. SELECT DISTINCT department\_code FROM Groups;
5. FROM — задаёт источник данных (таблицу, представление, подзапрос).
6. JOIN — объединяет данные из нескольких таблиц.  
   Пример:
7. SELECT s.student\_surname, g.group\_name
8. FROM Students s
9. JOIN Groups g ON s.group\_code = g.group\_code;
10. WHERE — фильтрация строк.
11. WHERE gpa > 3.5;
12. GROUP BY — группировка данных по заданным полям.
13. GROUP BY group\_code;
14. HAVING — фильтрация уже сгруппированных данных.
15. HAVING AVG(gpa) > 3.0;
16. ORDER BY — сортировка результата.

ORDER BY student\_surname ASC;