НАО «Карагандинский Технический Университет имени Абылкаса Сагинова»

Кафедра «Кибербезопасности и искусственного интеллекта»

Лабораторная работа №2

по дисциплине «Введение в базы данных»

Тема: Создание баз данных и таблиц в среде MS SQL Server.  
Информационное наполнение

Принял:

преп. Жакина М. М.

Выполнил:

студ. гр. СИБ-23-7 Бобер О.Н.

Караганда 2025

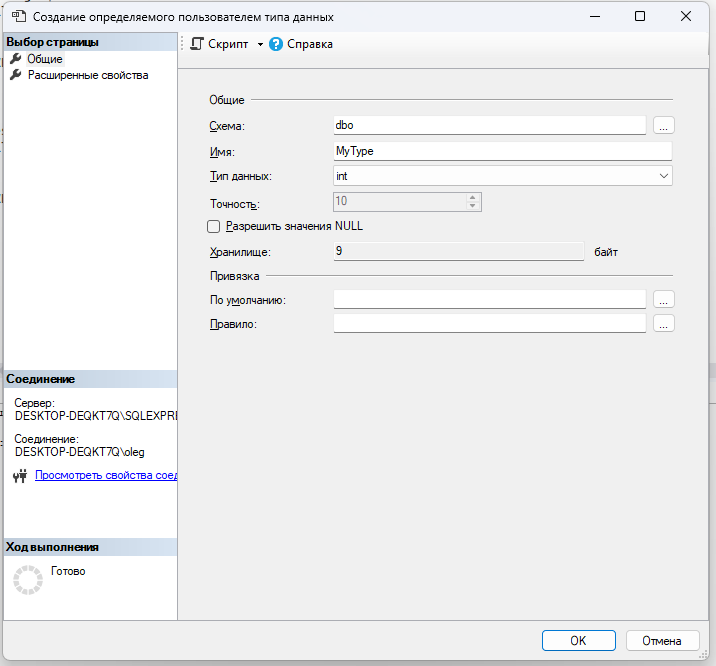
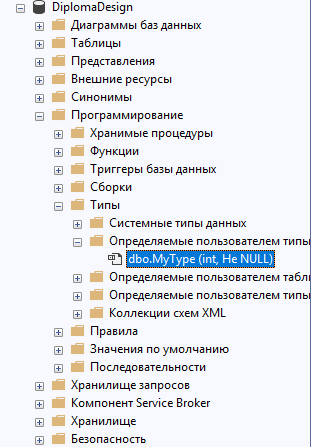
**Цель работы**

* усвоить способы создания таблиц, умолчаний, правил, ограничений БД средствами СУБД MS SQL Server;
* усвоить способы создания индексов, отношений и схемы отношений (диаграммы) базы данных средствами СУБД MS SQL Server;
* усвоить способы создания вода, удаления, редактирования данных в таблицах БД MS SQL Server.

**Содержание отчета**

1. Название и цель работы.
2. Индивидуальное задание.
3. Скрипты на создание пользовательского типа данных и таблиц БД, в соответствии с индивидуальным заданием.
4. Схема (диаграмма) БД с аннотацией.
5. Описание зависимостей между таблицами БД, включая правила ограничений целостности и созданные индексы.
6. Скрины экранной формы Edit Top 200 rows для каждой из таблиц БД с тестовыми наборами данных.

**Индивидуальное задание**

1. Создать пользовательский тип данных.  
     
     
   В обозревателе объектов:  
   
2. Определить таблицы базы данных в соответствии с требованиями индивидуального варианта задания (при создании таблиц определите в них первичные ключи и при необходимости используйте значения по умолчанию, ограничения и правила и созданный вами в п. 1 пользовательский тип данных).
3. Построить диаграмму вашей базы данных.
4. Определить внешние ключи.
5. Создать неуникальные индексы для атрибутов таблиц базы данных, которые являются внешними ключами.
6. Построить отношения (связи) между таблицами базы данных и задать правила ограничений целостности.
7. Ввести в ваши таблицы тестовые наборы данных (в родительские таблицы – не менее 10 записей, в дочерние – не менее 20). При необходимости откорректируйте созданные в лабораторной работе № 2 ограничения на вводимые данные.

Создание таблиц:

USE DiplomaDesign

GO

CREATE TABLE Departments (

department\_code INT PRIMARY KEY,

department\_name NVARCHAR(60) NOT NULL

);

CREATE TABLE Specialties (

specialty\_code INT PRIMARY KEY,

specialty\_name NVARCHAR(60) NOT NULL

);

CREATE TABLE Groups (

group\_code INT PRIMARY KEY,

group\_name NVARCHAR(60) NOT NULL,

department\_code INT NOT NULL,

FOREIGN KEY (department\_code) REFERENCES Departments(department\_code)

);

CREATE TABLE Students (

student\_code INT PRIMARY KEY,

student\_surname NVARCHAR(60) NOT NULL,

student\_first\_name NVARCHAR(60) NOT NULL,

student\_patronymic NVARCHAR(60) NULL,

group\_code INT NOT NULL,

specialty\_code INT NOT NULL,

gpa FLOAT DEFAULT 0.0 CHECK (gpa BETWEEN 0 AND 4),

FOREIGN KEY (group\_code) REFERENCES Groups(group\_code),

FOREIGN KEY (specialty\_code) REFERENCES Specialties(specialty\_code)

);

CREATE TABLE Professors (

professor\_code INT PRIMARY KEY,

professor\_surname NVARCHAR(60) NOT NULL,

professor\_first\_name NVARCHAR(60) NOT NULL,

professor\_patronymic NVARCHAR(60) NULL,

department\_code INT NOT NULL,

graduates\_number INT DEFAULT 0 CHECK (graduates\_number >= 0),

vacancies\_number INT DEFAULT 0 CHECK (vacancies\_number >= 0),

FOREIGN KEY (department\_code) REFERENCES Departments(department\_code)

);

CREATE TABLE GAK\_Composition (

specialty\_code INT NOT NULL,

iin CHAR(12) NOT NULL,

functions NVARCHAR(60) NOT NULL

CHECK (functions IN (N'председатель', N'секретарь', N'член ГАКа')),

PRIMARY KEY (specialty\_code, iin),

FOREIGN KEY (specialty\_code) REFERENCES Specialties(specialty\_code)

);

CREATE TABLE Projects\_Categories\_Types (

project\_category\_type\_code INT PRIMARY KEY,

project\_category\_type\_name NVARCHAR(60) NOT NULL

);

CREATE TABLE Banks (

bank\_code INT PRIMARY KEY,

bank\_name NVARCHAR(60) NOT NULL

);

CREATE TABLE GAK\_Reviewers (

iin CHAR(12) PRIMARY KEY,

reviewer\_surname NVARCHAR(60) NOT NULL,

reviewer\_first\_name NVARCHAR(60) NOT NULL,

reviewer\_patronymic NVARCHAR(60) NULL,

bank\_code INT NULL,

home\_address NVARCHAR(60) NULL,

phone\_number NVARCHAR(30) NULL,

work\_place NVARCHAR(60) NULL,

job\_title NVARCHAR(60) NULL,

identity\_card\_number INT NOT NULL,

document\_issue\_date DATE NOT NULL,

who\_issued\_document NVARCHAR(60) DEFAULT N'Не указано',

graduates\_number INT DEFAULT 0 CHECK (graduates\_number >= 0),

FOREIGN KEY (bank\_code) REFERENCES Banks(bank\_code)

);

CREATE TABLE Projects (

student\_code INT NOT NULL,

professor\_code INT NOT NULL,

graduation\_project\_topic NVARCHAR(60) NOT NULL,

project\_category\_type\_code INT NOT NULL,

reviewer\_code CHAR(12) NULL,

PRIMARY KEY (student\_code),

FOREIGN KEY (student\_code) REFERENCES Students(student\_code),

FOREIGN KEY (professor\_code) REFERENCES Professors(professor\_code),

FOREIGN KEY (project\_category\_type\_code) REFERENCES Projects\_Categories\_Types(project\_category\_type\_code),

FOREIGN KEY (reviewer\_code) REFERENCES GAK\_Reviewers(iin)

);

CREATE TABLE Projects\_Defense\_Schedule (

defense\_date DATE NOT NULL,

student\_code INT NOT NULL,

PRIMARY KEY (defense\_date, student\_code),

FOREIGN KEY (student\_code) REFERENCES Students(student\_code)

);

CREATE TABLE GAK\_Results (

result\_id INT PRIMARY KEY IDENTITY,

defense\_date DATE NOT NULL,

student\_code INT NOT NULL,

grade INT NOT NULL CHECK (grade BETWEEN 0 AND 100),

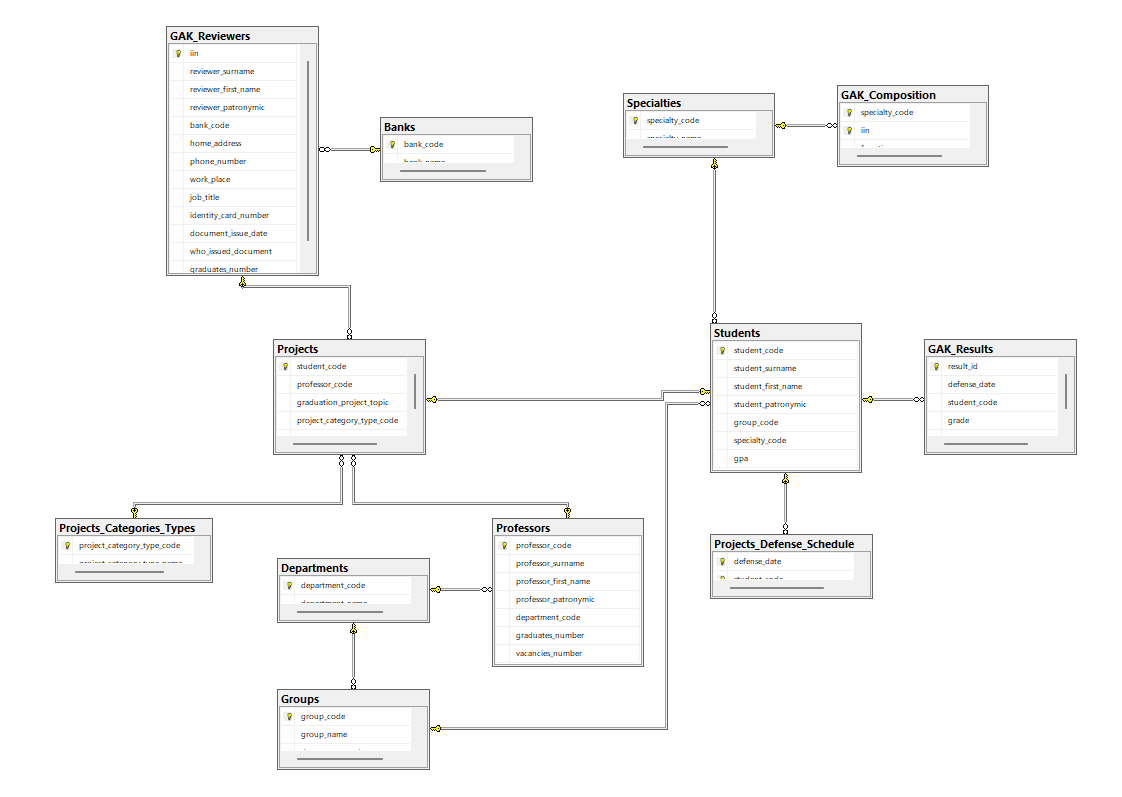
final\_assessment\_type NVARCHAR(60) NOT NULL

CHECK (final\_assessment\_type IN (N'гос. экзамен', N'защита дипл. проекта')),

FOREIGN KEY (student\_code) REFERENCES Students(student\_code)

);

Диаграмма:



Индексы:

USE DiplomaDesign;

GO

CREATE INDEX IX\_Groups\_DepartmentCode

ON Groups (department\_code);

CREATE INDEX IX\_Students\_GroupCode

ON Students (group\_code);

CREATE INDEX IX\_Students\_SpecialtyCode

ON Students (specialty\_code);

CREATE INDEX IX\_Professors\_DepartmentCode

ON Professors (department\_code);

CREATE INDEX IX\_GAK\_Composition\_SpecialtyCode

ON GAK\_Composition (specialty\_code);

CREATE INDEX IX\_GAK\_Reviewers\_BankCode

ON GAK\_Reviewers (bank\_code);

CREATE INDEX IX\_Projects\_StudentCode

ON Projects (student\_code);

CREATE INDEX IX\_Projects\_ProfessorCode

ON Projects (professor\_code);

CREATE INDEX IX\_Projects\_ProjectCategoryTypeCode

ON Projects (project\_category\_type\_code);

CREATE INDEX IX\_Projects\_ReviewerCode

ON Projects (reviewer\_code);

CREATE INDEX IX\_Projects\_Defense\_Schedule\_StudentCode

ON Projects\_Defense\_Schedule (student\_code);

CREATE INDEX IX\_GAK\_Results\_DefenseDate\_StudentCode

ON GAK\_Results (defense\_date, student\_code);

Заполнение данными в файле SQL.

**Контрольные вопросы**

1. Перечень поддерживаемых типов данных

SQL Server поддерживает:

Числовые: INT, BIGINT, SMALLINT, TINYINT, DECIMAL, NUMERIC, FLOAT, REAL, MONEY.

Символьные: CHAR, VARCHAR, NCHAR, NVARCHAR, TEXT, NTEXT.

Дата и время: DATE, TIME, DATETIME, SMALLDATETIME, DATETIME2.

Логические: BIT.

Двоичные: BINARY, VARBINARY, IMAGE.

Прочие: UNIQUEIDENTIFIER, XML, GEOGRAPHY, GEOMETRY.

2. Пользовательский тип данных

Это тип, созданный пользователем на основе встроенного.

Используется для унификации форматов полей (например, ИИН, телефон, код).

Создаётся:

CREATE TYPE IINType FROM CHAR(12) NOT NULL;

Удаляется:

DROP TYPE IINType;

3. Способы определения таблиц

Через SQL-команду:

CREATE TABLE имя\_таблицы (...);

Через графический интерфейс SQL Server Management Studio (SSMS) — Design Table.

Импорт из других источников (CSV, Excel, скрипт).

4. Назначение умолчаний, правил и ограничений

Умолчание (DEFAULT): задаёт значение поля по умолчанию, если оно не указано при вставке.

Правило (RULE): задаёт выражение, определяющее допустимые значения.

Ограничение (CONSTRAINT): обеспечивает целостность данных (например, уникальность, внешний ключ).

5. Перечень поддерживаемых умолчаний, ограничений, правил

DEFAULT — значение по умолчанию.

CHECK — проверка допустимости значений.

NOT NULL — запрет пустых значений.

PRIMARY KEY — уникальный идентификатор записи.

FOREIGN KEY — связь с другой таблицей.

UNIQUE — уникальные значения.

RULE — отдельное правило, применяемое к полю (устаревший механизм).

6. Способы назначения умолчаний, ограничений, правил

При создании таблицы:

CREATE TABLE Students (

gpa FLOAT DEFAULT 0 CHECK (gpa BETWEEN 0 AND 4)

);

Через ALTER TABLE:

ALTER TABLE Students

ADD CONSTRAINT CK\_GPA CHECK (gpa BETWEEN 0 AND 4);

7. Что представляют собой Null-значение

NULL — это отсутствие данных, не "ноль" и не "пустая строка".

Означает: значение неизвестно, не определено, отсутствует.

8. Свойство колонки IDENTITY

IDENTITY — автоматически увеличивающееся поле (аналог автоинкремента).

Пример:

student\_code INT IDENTITY(1,1) PRIMARY KEY

Где (1,1) — стартовое значение и шаг инкремента.

9. Назначение механизма отношений. Виды отношений

Отношения обеспечивают целостность и связанность данных между таблицами.

Виды:

Один-к-одному (1:1)

Один-ко-многим (1:М) — наиболее распространённый.

Многие-ко-многим (М:М) — реализуется через промежуточную таблицу.

10. Способы создания отношений

Через FOREIGN KEY в SQL:

FOREIGN KEY (department\_code) REFERENCES Departments(department\_code)

Через диаграмму связей (Database Diagram) в SSMS.

11. Ссылочная целостность данных. Способы её поддержания

Ссылочная целостность — это гарантия, что внешний ключ всегда указывает на существующую запись.

Поддерживается:

Ограничением FOREIGN KEY.

Опциями ON DELETE CASCADE, ON UPDATE CASCADE.

12. Способы отображения зависимостей между таблицами

Диаграмма базы данных (Database Diagram) в SSMS.

Системные представления:

INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

sys.foreign\_keys

Диаграммы в сторонних ERD-инструментах (dbdiagram.io, draw.io).

13. Способы построения и изменения схемы БД

Создание: CREATE SCHEMA имя\_схемы

Изменение: ALTER SCHEMA

Перемещение объектов между схемами:

ALTER SCHEMA Accounting TRANSFER dbo.Departments;

Также через SSMS → Database Diagrams.

14. Назначение и классификация индексов

Индексы ускоряют поиск и сортировку данных.

Классификация:

По уникальности: UNIQUE, NON-UNIQUE.

По структуре: кластерный (CLUSTERED), некластерный (NONCLUSTERED).

По количеству столбцов: одно- и многостолбцовые.

По включённым столбцам (INCLUDE).

15. Способы создания индексов

-- Кластерный (обычно на PK)

CREATE CLUSTERED INDEX IX\_Students\_Code ON Students(student\_code);

-- Некластерный (по FK)

CREATE NONCLUSTERED INDEX IX\_Students\_GroupCode ON Students(group\_code);

Также через SSMS → Tables → Indexes → New Index.

16. Способы ввода данных в таблицы. Ограничения целостности при вводе данных

Ввод вручную (INSERT INTO ... VALUES).

Импорт из файлов (CSV, Excel).

Через приложения (формы, скрипты).

Ограничения целостности (PK, FK, CHECK, NOT NULL) автоматически проверяются при вставке. Если нарушены — вставка отклоняется.

17. Понятие транзакции. Свойства транзакций

Транзакция — логическая группа операций, выполняемая как единое целое.

Свойства (ACID):

A — Atomicity: неделимость.

C — Consistency: согласованность.

I — Isolation: изолированность.

D — Durability: устойчивость после фиксации (COMMIT).

18. Удаление данных, усечение таблиц, удаление таблиц

Удаление строк:

DELETE FROM Students WHERE gpa < 2.0;

Полная очистка (без условий):

TRUNCATE TABLE Students;

Удаление таблицы:

DROP TABLE Students;