**Лабораторная работа № 2**

Тема: **"** **Создание баз данных и таблиц в среде MS SQL Server.**

**Информационное наполнение"**

**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.

**Лабораторная работа** рассчитана на 3 часа аудиторных занятий и состоит в изучении теоретического материала и получении практических навыков по определению таблиц базы данных, созданию правил, ограничений, умолчаний, пользовательских типов данных, индексов. Сдача лабораторной работы заключается в ответе на контрольные вопросы и демонстрации индивидуального задания на ПК.

**Содержание отчета:**

1. **Индивидуальное задание**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Подсистема «Расписание занятий»:   * *Кафедры* (код кафедры, наименование); * *Группа* (код группы, наименование, код кафедры, численность группы); * *Преподаватели* (код преподавателя, ФИО, код кафедры); * *Предметы* (Код предмета, наименование); * *Период проведения занятий* (Номер занятий, время начала, время окончания); * *Расписание* (день недели, номер занятия, код группы, код предмета, код типа занятия, номер ауд., код преподавателя); * *Тип занятий/аудиторий* (Код, наименование[\*\*\* может принимать следующие значения:лекции, практика, компьют, лабор..]); * *Аудитории* (Номер ауд, код типа занятий,  количество посадочных мест); * *Занятость\_аудиторий* (Код ауд, номер занятия,  день недели, признак занятости); * *Занятость\_препод* (Код преподавателя, номер занятия, признак занятости, день недели);   *Занятость группы* (Код группы, номер занятия, признак занятости, день недели); | 1. У каких преподавателей  можно прослушать курс лекций по «*i-ой*» дисциплине 2. Список незадействованных аудиторий по дням на *I-ом* занятии 3. Количество занятий у каждого из преподавателей по дням недели 4. Вычислить  количество свободных посадочных мест на парах в аудиториях по понедельникам 5. Выдать список предметов в наименовании которых встречается слово «…», читаемых преподавателями «*I-ой*» кафедры | 1   Расписание занятий для студентов *I-ой* группы  2  Расписание занятий «*I-го*» преподавателя  3  Список кафедр, преподаватели которых проводят занятия в компьютерных аудиториях | 1. Из таблицы *Аудитории* выбрать строки по условию: количество посадочных мест>50; 2. Из таблицы *Занятость группы* выбрать строки по условию: код  «*I-ой*» группы и  «*j-ый*» номер занятия (\* код группы и номер занятия задавать как параметр) 3. Вставить три новых строки  (с использованием цикла) в таблицу *Предметы* 4. Вычислить длительность занятий в минутах из таблицы *Период проведения пар* (\* номер занятия задавать как параметр) |

1. Скрипты на создание  пользовательского типа данных и таблиц БД, в соответствии с индивидуальным заданием.

-- БЛОК 1: СОЗДАНИЕ БАЗЫ ДАННЫХ

-- Лабораторная работа №2

-- Создание базы данных "Расписание\_занятий" и таблиц

-- Создаем базу если нет

IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = 'Расписание\_занятий')

CREATE DATABASE Расписание\_занятий

GO

USE Расписание\_занятий

GO

-- =============================================

-- БЛОК 4: СОЗДАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ

-- =============================================

-- Создаем таблицы если их нет

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Кафедры' AND type = 'U')

CREATE TABLE Кафедры (

Код\_кафедры INT NOT NULL PRIMARY KEY,

Наименование NVARCHAR(100) NOT NULL,

Дата\_создания DATE DEFAULT GETDATE()

)

GO

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Преподаватели' AND type = 'U')

CREATE TABLE Преподаватели (

Код\_преподавателя INT NOT NULL PRIMARY KEY,

ФИО NVARCHAR(100) NOT NULL,

Должность NVARCHAR(50) DEFAULT 'Преподаватель',

Код\_кафедры INT,

Ставка DECIMAL(3,2) DEFAULT 1.0 CHECK (Ставка BETWEEN 0.25 AND 2.0)

)

GO

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Группы' AND type = 'U')

CREATE TABLE Группы (

Код\_группы INT NOT NULL PRIMARY KEY,

Наименование NVARCHAR(50) NOT NULL UNIQUE,

Код\_кафедры INT,

Курс TINYINT DEFAULT 1 CHECK (Курс BETWEEN 1 AND 6)

)

GO

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Предметы' AND type = 'U')

CREATE TABLE Предметы (

Код\_предмета INT NOT NULL PRIMARY KEY,

Наименование NVARCHAR(100) NOT NULL,

Код\_кафедры INT,

Часы\_лекций SMALLINT DEFAULT 0 CHECK (Часы\_лекций >= 0),

Часы\_практик SMALLINT DEFAULT 0 CHECK (Часы\_практик >= 0)

)

GO

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Тип\_занятий' AND type = 'U')

CREATE TABLE Тип\_занятий (

Код\_типа INT NOT NULL PRIMARY KEY,

Наименование NVARCHAR(20) NOT NULL CHECK (Наименование IN ('Лекция', 'Практика', 'Лабораторная', 'Семинар'))

)

GO

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Период\_занятий' AND type = 'U')

CREATE TABLE Период\_занятий (

Код\_периода INT NOT NULL PRIMARY KEY,

Наименование NVARCHAR(10) NOT NULL,

Время\_начала TIME NOT NULL,

Время\_окончания TIME NOT NULL

)

GO

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Аудитории' AND type = 'U')

CREATE TABLE Аудитории (

Код\_аудитории INT NOT NULL PRIMARY KEY,

Номер\_аудитории NVARCHAR(10) NOT NULL,

Тип\_аудитории NVARCHAR(20) DEFAULT 'Учебная' CHECK (Тип\_аудитории IN ('Учебная', 'Лекционная', 'Лаборатория', 'Компьютерный класс')),

Вместимость SMALLINT DEFAULT 20 CHECK (Вместимость BETWEEN 10 AND 300)

)

GO

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Расписание' AND type = 'U')

CREATE TABLE Расписание (

Код\_занятия INT IDENTITY(1,1) PRIMARY KEY,

Дата\_занятия DATE NOT NULL,

Код\_периода INT NOT NULL,

Код\_группы INT NOT NULL,

Код\_предмета INT NOT NULL,

Код\_преподавателя INT NOT NULL,

Код\_аудитории INT NOT NULL,

Код\_типа INT NOT NULL

)

GO

-- =============================================

-- БЛОК 5: СОЗДАНИЕ ВНЕШНИХ КЛЮЧЕЙ И ОТНОШЕНИЙ

-- =============================================

-- Внешние ключи для таблицы Преподаватели

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Преподаватели\_Кафедры')

BEGIN

ALTER TABLE Преподаватели

ADD CONSTRAINT FK\_Преподаватели\_Кафедры

FOREIGN KEY (Код\_кафедры) REFERENCES Кафедры(Код\_кафедры)

ON DELETE SET NULL

END

GO

-- Внешние ключи для таблицы Группы

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Группы\_Кафедры')

BEGIN

ALTER TABLE Группы

ADD CONSTRAINT FK\_Группы\_Кафедры

FOREIGN KEY (Код\_кафедры) REFERENCES Кафедры(Код\_кафедры)

ON DELETE SET NULL

END

GO

-- Внешние ключи для таблицы Предметы

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Предметы\_Кафедры')

BEGIN

ALTER TABLE Предметы

ADD CONSTRAINT FK\_Предметы\_Кафедры

FOREIGN KEY (Код\_кафедры) REFERENCES Кафедры(Код\_кафедры)

ON DELETE SET NULL

END

GO

-- Внешние ключи для таблицы Расписание

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Расписание\_Период')

BEGIN

ALTER TABLE Расписание

ADD CONSTRAINT FK\_Расписание\_Период

FOREIGN KEY (Код\_периода) REFERENCES Период\_занятий(Код\_периода)

ON DELETE CASCADE

END

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Расписание\_Группы')

BEGIN

ALTER TABLE Расписание

ADD CONSTRAINT FK\_Расписание\_Группы

FOREIGN KEY (Код\_группы) REFERENCES Группы(Код\_группы)

ON DELETE CASCADE

END

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Расписание\_Предметы')

BEGIN

ALTER TABLE Расписание

ADD CONSTRAINT FK\_Расписание\_Предметы

FOREIGN KEY (Код\_предмета) REFERENCES Предметы(Код\_предмета)

ON DELETE CASCADE

END

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Расписание\_Преподаватели')

BEGIN

ALTER TABLE Расписание

ADD CONSTRAINT FK\_Расписание\_Преподаватели

FOREIGN KEY (Код\_преподавателя) REFERENCES Преподаватели(Код\_преподавателя)

ON DELETE CASCADE

END

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Расписание\_Аудитории')

BEGIN

ALTER TABLE Расписание

ADD CONSTRAINT FK\_Расписание\_Аудитории

FOREIGN KEY (Код\_аудитории) REFERENCES Аудитории(Код\_аудитории)

ON DELETE CASCADE

END

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE name = 'FK\_Расписание\_Тип')

BEGIN

ALTER TABLE Расписание

ADD CONSTRAINT FK\_Расписание\_Тип

FOREIGN KEY (Код\_типа) REFERENCES Тип\_занятий(Код\_типа)

ON DELETE CASCADE

END

GO

-- =============================================

-- БЛОК 6: СОЗДАНИЕ ИНДЕКСОВ ДЛЯ ВНЕШНИХ КЛЮЧЕЙ

-- =============================================

-- СОЗДАНИЕ НЕУНИКАЛЬНЫХ ИНДЕКСОВ ДЛЯ ВНЕШНИХ КЛЮЧЕЙ

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Расписание\_Период' AND object\_id = OBJECT\_ID('Расписание'))

CREATE NONCLUSTERED INDEX IX\_Расписание\_Период ON Расписание(Код\_периода)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Расписание\_Группы' AND object\_id = OBJECT\_ID('Расписание'))

CREATE NONCLUSTERED INDEX IX\_Расписание\_Группы ON Расписание(Код\_группы)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Расписание\_Предметы' AND object\_id = OBJECT\_ID('Расписание'))

CREATE NONCLUSTERED INDEX IX\_Расписание\_Предметы ON Расписание(Код\_предмета)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Расписание\_Преподаватели' AND object\_id = OBJECT\_ID('Расписание'))

CREATE NONCLUSTERED INDEX IX\_Расписание\_Преподаватели ON Расписание(Код\_преподавателя)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Расписание\_Аудитории' AND object\_id = OBJECT\_ID('Расписание'))

CREATE NONCLUSTERED INDEX IX\_Расписание\_Аудитории ON Расписание(Код\_аудитории)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Расписание\_Тип' AND object\_id = OBJECT\_ID('Расписание'))

CREATE NONCLUSTERED INDEX IX\_Расписание\_Тип ON Расписание(Код\_типа)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Преподаватели\_Кафедры' AND object\_id = OBJECT\_ID('Преподаватели'))

CREATE NONCLUSTERED INDEX IX\_Преподаватели\_Кафедры ON Преподаватели(Код\_кафедры)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Группы\_Кафедры' AND object\_id = OBJECT\_ID('Группы'))

CREATE NONCLUSTERED INDEX IX\_Группы\_Кафедры ON Группы(Код\_кафедры)

GO

IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE name = 'IX\_Предметы\_Кафедры' AND object\_id = OBJECT\_ID('Предметы'))

CREATE NONCLUSTERED INDEX IX\_Предметы\_Кафедры ON Предметы(Код\_кафедры)

GO

-- =============================================

-- БЛОК 7: ЗАПОЛНЕНИЕ ТАБЛИЦ ТЕСТОВЫМИ ДАННЫМИ

-- =============================================

-- Вставляем данные только если таблицы пустые

IF NOT EXISTS (SELECT 1 FROM Кафедры)

BEGIN

INSERT INTO Кафедры (Код\_кафедры, Наименование, Дата\_создания) VALUES

(1, 'Кафедра информатики и вычислительной техники', '2000-09-01'),

(2, 'Кафедра математики', '1998-08-15'),

(3, 'Кафедра физики', '1995-07-20'),

(4, 'Кафедра иностранных языков', '2005-03-10'),

(5, 'Кафедра экономики', '2008-11-25'),

(6, 'Кафедра программирования', '2010-01-30'),

(7, 'Кафедра сетевых технологий', '2012-06-18'),

(8, 'Кафедра базы данных', '2013-09-05'),

(9, 'Кафедра веб-разработки', '2014-02-14'),

(10, 'Кафедра искусственного интеллекта', '2015-10-08'),

(11, 'Кафедра кибербезопасности', '2018-12-12')

END

GO

IF NOT EXISTS (SELECT 1 FROM Преподаватели)

BEGIN

INSERT INTO Преподаватели (Код\_преподавателя, ФИО, Должность, Код\_кафедры, Ставка) VALUES

(1, 'Иванов А.С.', 'Профессор', 1, 1.5),

(2, 'Петрова М.И.', 'Доцент', 1, 1.0),

(3, 'Сидоров В.П.', 'Преподаватель', 2, 0.75),

(4, 'Козлова Е.В.', 'Доцент', 3, 1.0),

(5, 'Николаев Д.С.', 'Профессор', 4, 1.5),

(6, 'Федорова О.П.', 'Преподаватель', 5, 1.0),

(7, 'Морозов И.А.', 'Доцент', 6, 1.0),

(8, 'Волкова Т.М.', 'Преподаватель', 7, 0.5),

(9, 'Белов С.К.', 'Профессор', 8, 1.5),

(10, 'Григорьева Л.Н.', 'Доцент', 9, 1.0)

END

GO

IF NOT EXISTS (SELECT 1 FROM Группы)

BEGIN

INSERT INTO Группы (Код\_группы, Наименование, Код\_кафедры, Курс) VALUES

-- 4 курс (выпуск 2025)

(1, 'ИВТ-22-1', 1, 4),

(2, 'ИВТ-22-2', 1, 4),

(3, 'ПИ-22-1', 6, 4),

(4, 'ПИ-22-2', 6, 4),

(5, 'КБ-22-1', 11, 4),

-- 3 курс (выпуск 2026)

(6, 'ИВТ-23-1', 1, 3),

(7, 'ИВТ-23-2', 1, 3),

(8, 'ПИ-23-1', 6, 3),

(9, 'ПИ-23-2', 6, 3),

(10, 'КБ-23-1', 11, 3),

-- 2 курс (выпуск 2027)

(11, 'ИВТ-24-1', 1, 2),

(12, 'ИВТ-24-2', 1, 2),

-- 1 курс (выпуск 2028)

(13, 'ИВТ-25-1', 1, 1),

(14, 'ИВТ-25-2', 1, 1),

(15, 'ПИ-25-1', 6, 1),

(16, 'КБ-25-1', 11, 1)

END

GO

IF NOT EXISTS (SELECT 1 FROM Предметы)

BEGIN

INSERT INTO Предметы (Код\_предмета, Наименование, Код\_кафедры, Часы\_лекций, Часы\_практик) VALUES

(1, 'Базы данных', 8, 36, 36),

(2, 'Программирование', 6, 54, 54),

(3, 'Математический анализ', 2, 72, 36),

(4, 'Физика', 3, 54, 36),

(5, 'Английский язык', 4, 36, 36),

(6, 'Сетевые технологии', 7, 36, 36),

(7, 'Веб-разработка', 9, 36, 54),

(8, 'Искусственный интеллект', 10, 36, 36),

(9, 'Кибербезопасность', 11, 36, 36),

(10, 'Экономика', 5, 36, 18)

END

GO

IF NOT EXISTS (SELECT 1 FROM Тип\_занятий)

BEGIN

INSERT INTO Тип\_занятий (Код\_типа, Наименование) VALUES

(1, 'Лекция'),

(2, 'Практика'),

(3, 'Лабораторная'),

(4, 'Семинар')

END

GO

IF NOT EXISTS (SELECT 1 FROM Период\_занятий)

BEGIN

INSERT INTO Период\_занятий (Код\_периода, Наименование, Время\_начала, Время\_окончания) VALUES

(1, '1 пара', '08:30', '10:00'),

(2, '2 пара', '10:10', '11:40'),

(3, '3 пара', '12:10', '13:40'),

(4, '4 пара', '14:00', '15:30'),

(5, '5 пара', '15:40', '17:10'),

(6, '6 пара', '17:20', '18:50')

END

GO

IF NOT EXISTS (SELECT 1 FROM Аудитории)

BEGIN

INSERT INTO Аудитории (Код\_аудитории, Номер\_аудитории, Тип\_аудитории, Вместимость) VALUES

(1, '101', 'Лекционная', 150),

(2, '102', 'Лекционная', 120),

(3, '201', 'Учебная', 30),

(4, '202', 'Учебная', 25),

(5, '301', 'Компьютерный класс', 20),

(6, '302', 'Компьютерный класс', 20),

(7, '401', 'Лаборатория', 15),

(8, '402', 'Лаборатория', 15),

(9, '501', 'Учебная', 35),

(10, '502', 'Учебная', 30)

END

GO

IF NOT EXISTS (SELECT 1 FROM Расписание)

BEGIN

INSERT INTO Расписание (Дата\_занятия, Код\_периода, Код\_группы, Код\_предмета, Код\_преподавателя, Код\_аудитории, Код\_типа) VALUES

-- Занятия для групп 2022-2025 годов

('2025-02-01', 1, 1, 1, 9, 5, 1),

('2025-02-01', 2, 1, 1, 9, 5, 3),

('2025-02-01', 3, 2, 1, 9, 6, 1),

('2025-02-01', 4, 2, 1, 9, 6, 3),

('2025-02-02', 1, 3, 2, 7, 5, 1),

('2025-02-02', 2, 3, 2, 7, 5, 2),

('2025-02-02', 3, 4, 2, 7, 6, 1),

('2025-02-02', 4, 4, 2, 7, 6, 2),

('2025-02-03', 1, 5, 9, 10, 7, 1),

('2025-02-03', 2, 5, 9, 10, 7, 3),

('2025-02-03', 3, 6, 3, 3, 1, 1),

('2025-02-03', 4, 6, 3, 3, 3, 2),

('2025-02-04', 1, 7, 4, 4, 2, 1),

('2025-02-04', 2, 7, 4, 4, 4, 3),

('2025-02-04', 3, 8, 7, 10, 5, 1),

('2025-02-04', 4, 8, 7, 10, 5, 3),

('2025-02-05', 1, 9, 6, 8, 6, 1),

('2025-02-05', 2, 9, 6, 8, 6, 3),

('2025-02-05', 3, 10, 8, 10, 7, 1),

('2025-02-05', 4, 10, 8, 10, 7, 3),

('2025-02-06', 1, 11, 5, 5, 9, 1),

('2025-02-06', 2, 11, 5, 5, 9, 2),

('2025-02-06', 3, 12, 10, 6, 10, 1),

('2025-02-06', 4, 12, 10, 6, 10, 2),

-- Дополнительные занятия для новых групп

('2025-02-07', 1, 13, 2, 7, 5, 1),

('2025-02-07', 2, 14, 2, 7, 6, 1),

('2025-02-07', 3, 15, 1, 9, 5, 1),

('2025-02-07', 4, 16, 9, 10, 7, 1)

END

GO

-- =============================================

-- БЛОК 8: ПРОВЕРКА РЕЗУЛЬТАТОВ

-- =============================================

-- Вывод информации о созданных объектах

SELECT 'База данных "Расписание\_занятий" создана успешно!' as Результат

SELECT COUNT(\*) as 'Количество таблиц' FROM sys.tables WHERE type = 'U'

SELECT name as 'Созданные таблицы' FROM sys.tables WHERE type = 'U' ORDER BY name

-- Проверка данных во ВСЕХ таблицах

SELECT 'Кафедры:' as Таблица, COUNT(\*) as Количество\_записей FROM Кафедры

UNION ALL

SELECT 'Преподаватели:', COUNT(\*) FROM Преподаватели

UNION ALL

SELECT 'Группы:', COUNT(\*) FROM Группы

UNION ALL

SELECT 'Предметы:', COUNT(\*) FROM Предметы

UNION ALL

SELECT 'Тип\_занятий:', COUNT(\*) FROM Тип\_занятий

UNION ALL

SELECT 'Период\_занятий:', COUNT(\*) FROM Период\_занятий

UNION ALL

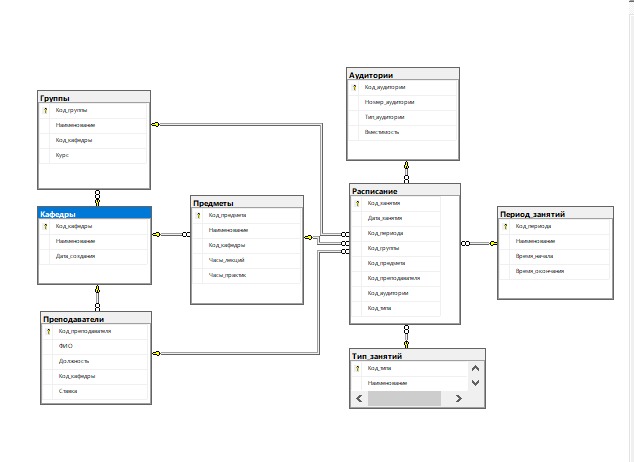
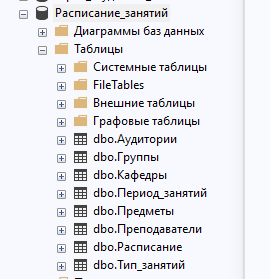
SELECT 'Аудитории:', COUNT(\*) FROM Аудитории

UNION ALL

SELECT 'Расписание:', COUNT(\*) FROM Расписание

GO

1. Схема (диаграмма) БД с аннотацией.



1. Описание зависимостей между таблицами БД, включая правила ограничений целостности и созданные индексы.

ОПИСАНИЕ ЗАВИСИМОСТЕЙ МЕЖДУ ТАБЛИЦАМИ БАЗЫ ДАННЫХ

1. СХЕМА ЗАВИСИМОСТЕЙ ТАБЛИЦ

КАФЕДРЫ (1) ← (M) ПРЕПОДАВАТЕЛИ

↑

└── (M) ГРУППЫ

↑

└── (M) ПРЕДМЕТЫ

↑

РАСПИСАНИЕ (M) → (1) ПЕРИОД\_ЗАНЯТИЙ

↑

└── (M) → (1) ТИП\_ЗАНЯТИЙ

↑

└── (M) → (1) АУДИТОРИИ

2. ПОДРОБНОЕ ОПИСАНИЕ ЗАВИСИМОСТЕЙ

2.1. КАФЕДРЫ → ПРЕПОДАВАТЕЛИ

- Связь: Один-ко-многим (1:M)

- Внешний ключ: `Преподаватели.Код\_кафедры → Кафедры.Код\_кафедры`

- Ограничения целостности:

- `ON UPDATE CASCADE` - при изменении кода кафедры обновляются преподаватели

- `ON DELETE NO ACTION` - запрет удаления кафедры с преподавателями

- Индекс: `IX\_Преподаватели\_Кафедры` на `Код\_кафедры`

2.2. КАФЕДРЫ → ГРУППЫ

- Связь Один-ко-многим (1:M)

- Внешний ключ: `Группы.Код\_кафедры → Кафедры.Код\_кафедры`

- Ограничения целостности:

- `ON UPDATE CASCADE` - каскадное обновление

- `ON DELETE NO ACTION` - запрет удаления кафедры с группами

- Индекс:\_Группы\_Кафедры` на `Код\_кафедры`

2.3 КАФЕДРЫ → ПРЕДМЕТЫ

- Связь: Один-ко-многим (1:M)

- Внешний ключ: `Предметы.Код\_кафедры → Кафедры.Код\_кафедры`

- Ограничения целостности:

- `ON UPDATE CASCADE` - каскадное обновление

- `ON DELETE NO ACTION` - запрет удаления кафедры с предметами

- Индекс:\_Предметы\_Кафедры` на `Код\_кафедры`

2.4. РАСПИСАНИЕ → ПЕРИОД\_ЗАНЯТИЙ

- Связь: Многие-к-одному (M:1)

- Внешний ключ: `Расписание.Код\_периода → Период\_занятий.Код\_периода`

- Ограничения целостности:

- `ON UPDATE CASCADE` - обновление при изменении периода

- `ON DELETE NO ACTION` - запрет удаления периода с занятиями

- Индекс:\_Расписание\_Период` на `Код\_периода`

2.5. РАСПИСАНИЕ → ГРУППЫ

- Связь: Многие-к-одному (M:1)

- Внешний ключ:`Расписание.Код\_группы → Группы.Код\_группы`

- Ограничения целостности:

- `ON UPDATE CASCADE` - обновление при изменении группы

- `ON DELETE CASCADE` - удаление занятий при удалении группы

-Индекс:Расписание\_Группы` на `Код\_группы`

2.6. РАСПИСАНИЕ → ПРЕДМЕТЫ

- Связь: Многие-к-одному (M:1)

- Внешний ключ:`Расписание.Код\_предмета → Предметы.Код\_предмета`

- Ограничения целостности:

- ON UPDATE CASCADE - обновление при изменении предмета

- ON DELETE NO ACTION - запрет удаления предмета с занятиями

-Индекс Расписание\_Предметы на Код\_предмета

2.7. РАСПИСАНИЕ → ПРЕПОДАВАТЕЛИ

- Связь: Многие-к-одному (M:1)

-Внешний ключ:`Расписание.Код\_преподавателя → Преподаватели.Код\_преподавателя`

- Ограничения целостности:

- ON UPDATE CASCADE - обновление при изменении преподавателя

- ON DELETE NO ACTION - запрет удаления преподавателя с занятиями

- Индекс: Расписание\_Преподаватели на Код\_преподавател

2.8. РАСПИСАНИЕ → АУДИТОРИИ

-Связь:Многие-к-одному (M:1)

- Внешний ключ:`Расписание.Код\_аудитории → Аудитории.Код\_аудитории

- Ограничения целостности:

- ON UPDATE CASCADE - обновление при изменении аудитории

- ON DELETE NO ACTION- запрет удаления аудитории с занятиями

- Индекс:\_Расписание\_Аудитории` на `Код\_аудитории`

2.9. РАСПИСАНИЕ → ТИП\_ЗАНЯТИЙ

- Связь: Многие-к-одному (M:1)

- Внешний ключ: `Расписание.Код\_типа → Тип\_занятий.Код\_типа`

- Ограничения целостности

- `ON UPDATE CASCADE` - обновление при изменении типа

- `ON DELETE NO ACTION` - запрет удаления типа с занятиями

- Индекс: `IX\_Расписание\_Тип` на `Код\_типа`

3. СОЗДАННЫЕ ИНДЕКСЫ

3.1. Уникальные индексы:

- UX\_Группы\_Наименование- уникальность наименований групп

- UQ\_Расписание\_Группа\_Время - предотвращение конфликтов групп

- UQ\_Расписание\_Аудитория\_Врем - предотвращение конфликтов аудиторий

- UQ\_Расписание\_Преподаватель\_Время - предотвращение конфликтов преподавателей

3.2. Неуникальные индексы для внешних ключей:

-Преподаватели\_Кафедры - поиск преподавателей по кафедре

- \*Группы\_Кафедры - поиск групп по кафедре

- Предметы\_Кафедры- поиск предметов по кафедре

- \_Расписание\_Период - фильтрация расписания по периоду

- \_Расписание\_Группы - поиск занятий группы

- \_Расписание\_Предметы - поиск занятий по предмету

- \_Расписание\_Преподаватели - расписание преподавателя

- \_Расписание\_Аудитории - занятость аудиторий

- Расписание\_Тип - фильтрация по типу занятия

4. ПРАВИЛА ОГРАНИЧЕНИЙ ЦЕЛОСТНОСТИ

4.1. Бизнес-правила:

1. Кафедра не может быть удалена если есть преподаватели, группы или предметы

2. Группа может быть удалена - все ее занятия удаляются автоматически

3. Преподаватель не может быть удален если ведет занятия

4. Предмет не может быть удале если есть занятия по нему

5. Аудитория не может быть удалена если есть занятия в ней

6. Тип занятия не может быть удален если используется в расписании

4.2. Ограничения CHECK:

- `Ставка BETWEEN 0.25 AND 2.0` - допустимый диапазон ставок

- `Курс BETWEEN 1 AND 6` - допустимые курсы обучения

- `Часы\_лекций >= 0`, `Часы\_практик >= 0` - неотрицательные часы

- `Вместимость BETWEEN 10 AND 300` - разумная вместимость аудиторий

#. РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ СКРИПТА

После выполнения скрипта в терминале появятся:

Таблицы:

- Кафедры

- Преподаватели

- Группы

- Предметы

- Тип\_занятий

- Период\_занятий

- Аудитории

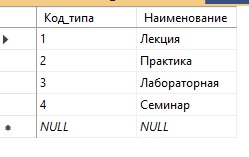
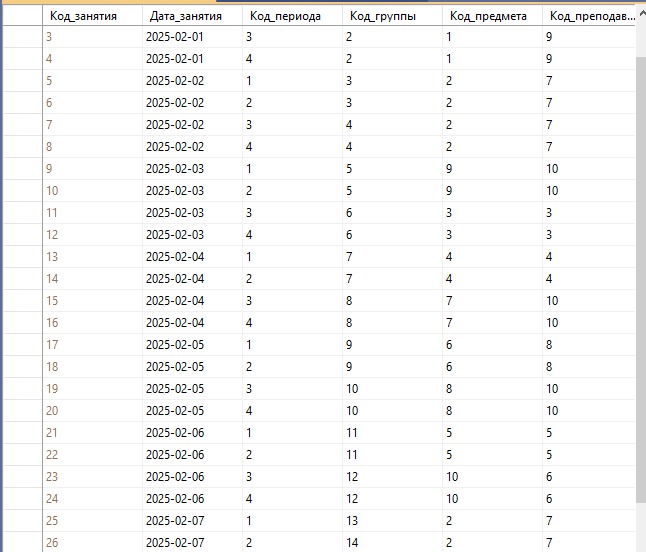
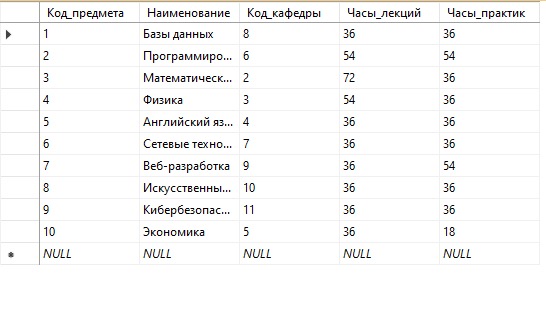
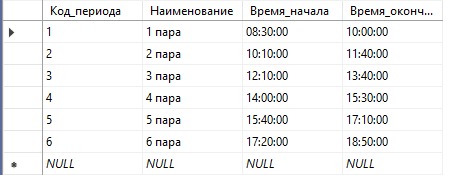
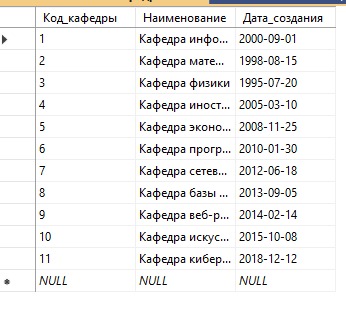
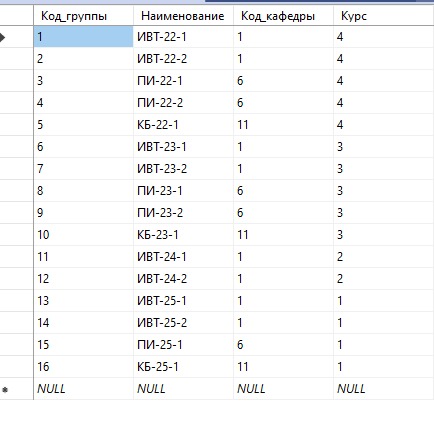
- Расписание

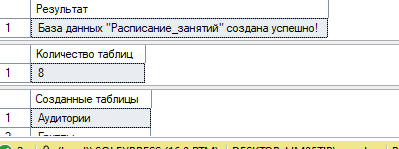
Ограничения: 15+ внешних ключей с настроенными действиями

Индексы:10+ индексов для оптимизации запросов

Данные: Тестовые данные для всех таблиц

1. Скрины экранной формы Edit Top 200 rows для каждой из таблиц БД с тестовыми наборами данных.





Контрольные вопросы по базам данных

1. Типы данных

- Числовые: INT, DECIMAL, FLOAT

- Строковые: VARCHAR, NVARCHAR, TEXT

- Дата/время: DATE, DATETIME

- Двоичные: BINARY, VARBINARY

- Логические: BIT

2. Пользовательский тип данных

Создание собственных типов для единообразия:

CREATE TYPE Phone FROM VARCHAR(20);

3. Способы определения таблиц

- CREATE TABLE

- SELECT INTO (из запроса)

- Графический интерфейс SSMS

4. Назначение ограничений

- DEFAULT - значение по умолчанию

- CHECK - проверка условий

- UNIQUE - уникальность

- PRIMARY KEY - первичный ключ

- FOREIGN KEY - внешний ключ

5. Способы назначения ограничений

При создании таблицы или через ALTER TABLE

6. NULL-значение

Специальное значение "отсутствие данных", не равно пустой строке или нулю

7. IDENTITY

Автоматическая нумерация строк:

ID INT IDENTITY(1,1) -- начало с 1, шаг 1

8. Отношения между таблицами

Виды:

- Один-ко-многим (1:M) - основной вид

- Один-к-одному (1:1) - редко

- Многие-ко-многим (M:M) - через связующую таблицу

9. Создание отношений

Через внешние ключи:

FOREIGN KEY (column) REFERENCES Table(PK)

10. Ссылочная целостность

Обеспечение корректности связей через:

- Внешние ключи

- Каскадное удаление/обновление

- Ограничения CHECK

11. Отображение зависимостей

- Диаграммы базы данных

- Системные представления

- Сторонние инструменты

12. Построение схемы БД

- DDL команды (CREATE, ALTER, DROP)

- Графические инструменты

- Миграционные скрипты

13. Индексы

Назначение: ускорение поиска

Классификация:

- Кластеризованные (порядок данных)

- Некластеризованные (отдельная структура)

- Уникальные, составные, фильтрованные

14. Создание индексов

CREATE INDEX IX\_Name ON Table(Column);

15. Ввод данных

Способы:

- INSERT INTO

- BULK INSERT

- Import Wizard

Ограничения: проверка типов, уникальность, внешние ключи

16. Транзакции

Логическая единица работы. Свойства ACID:

- Atomicity (Атомарность) - все или ничего

- Consistency (Согласованность) - целостность данных

- Isolation (Изолированность) - параллельные транзакции

- Durability (Долговечность) - сохранение результатов

17. Удаление данных

- DELETE - удаление строк (можно откатить)Ф

- TRUNCATE - быстрое удаление всех данных

- DROP TABLE - полное удаление таблицы