НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«МЭИ»

# Расчетное задание по дисциплине «Базы данных»

## на тему:

«Разработка базы данных кафедры»

Выполнил: студент группы А-1-19

Доржу Н.Ш. Руководитель: Зейн А.Н.

Москва 2021

Оглавление

[Введение 3](#_TOC_250018)

[Цель работы 3](#_TOC_250017)

[Предметная область 3](#_TOC_250016)

[Глава 1. Проектирование базы данных 4](#_TOC_250015)

* 1. [Инфологическая модель 4](#_TOC_250014)
  2. [Логическая модель 5](#_TOC_250013)

[Глава 2. Реализация БД 7](#_TOC_250012)

* 1. [СУБД Microsoft SQL Server 7](#_TOC_250011)
  2. [Описание физической модели БД 7](#_TOC_250010)

[Глава 3. Реализация базы данных 10](#_TOC_250009)

[3.1. Язык Высокого Уровня С#. Технология ADO .NET 10](#_TOC_250008)

[Заключение 11](#_TOC_250006)

# Введение

## Цель работы

Разработка базы данных кафедры. БД должна поддерживать выполнение следующих функций:

* поиск учебного плана (по семестрам) для каждого курса;
* поиск расписания занятий для преподавателей;
* расчет нагрузки для преподавателей;
* учет кафедрального плана изданий методической литературы;
* формирование отчета выполнения плана изданий (по преподавателям, по месяцам, по дисциплинам)
* списков студентов-дипломников (по преподавателям).

## Предметная область

База данных кафедры - учет учебного расписания, плана изданий и студентов-дипломников.

Обозначим правила:

* Нет разделения на потоки
* Преподаватель ведет как минимум один предмет. Предмет ведет один преподаватель
* Автор методической литературы один преподаватель

# Глава 1. Проектирование базы данных

В результате анализа решаемой задачи можно выделить следующие информационные объекты, данные о которых должны содержаться в проектируемой БД: преподаватель, группа, предмет, литература, занятия, студент-дипломник.

## Инфологическая модель

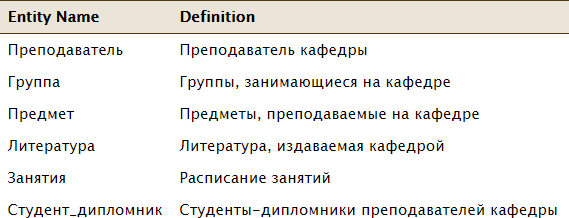
Описание сущностей БД получено с помощью генератора отчетов CASE-средства с.

Рис 1.1. Сущности

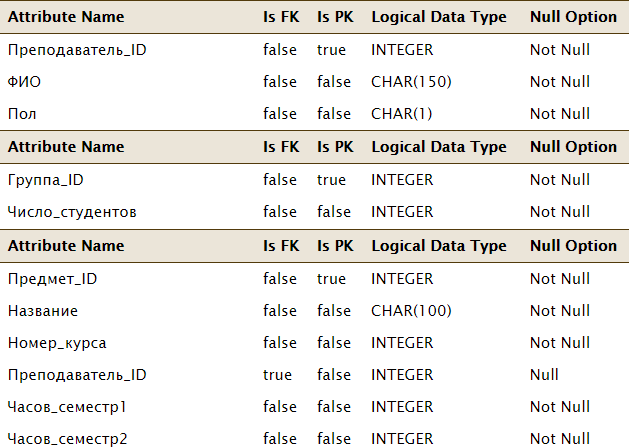


Рис 1.2. Атрибуты сущностей Преподаватель, Группа, Предмет

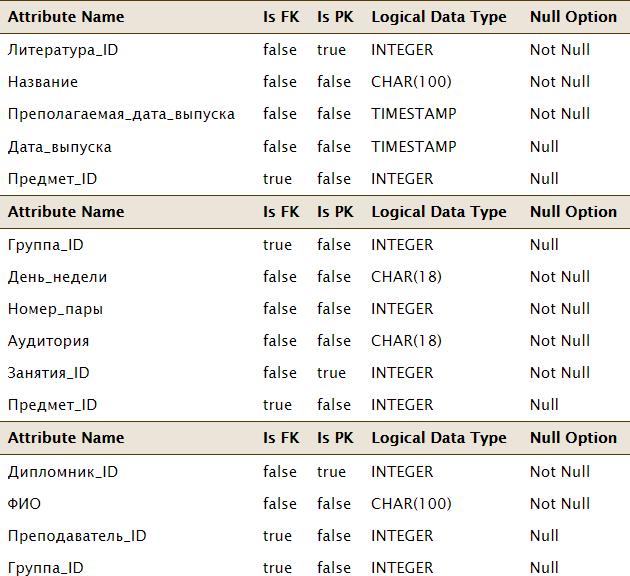


Рис 1.3. Атрибуты сущностей Литература, Занятия, Студент-дипломник

## Логическая модель

На основе построенной ER-модели для разработки БД создана реляционная модель.

Выполнено несколько этапов:

* + - Преобразования сущностей ER-модели в отношения РМД Выполнено в ERwin Data Modeler.
    - Приведение отношения к 3НФ

Покажем, модель (рис. 1.1) находится в 3НФ

1. РБД в 1НФ, так как каждое поле таблицы содержит атомарное значение.
2. РБД в 2НФ, так как:
   1. РБД в 1НФ.
   2. Каждый неключевой атрибут таблицы функционально зависит от ее ключа. Все отношения имеют простые ключи, таким образом, все атрибуты функционально зависят от ключа и не зависят ни от одного подмножества.
3. РБД в 3НФ:
   1. РБД в 2НФ.
   2. Нет транзитивных зависимостей (неключевые атрибуты не зависят от других неключевых атрибутов)
      * Определены правила целостности для отношений РМД

В результате выполнения указанных этапов была получена реляционная модель (рис. 1.1). **Описание отношений, их атрибутов, ключей и правил целостности приведены в Приложении 1**.

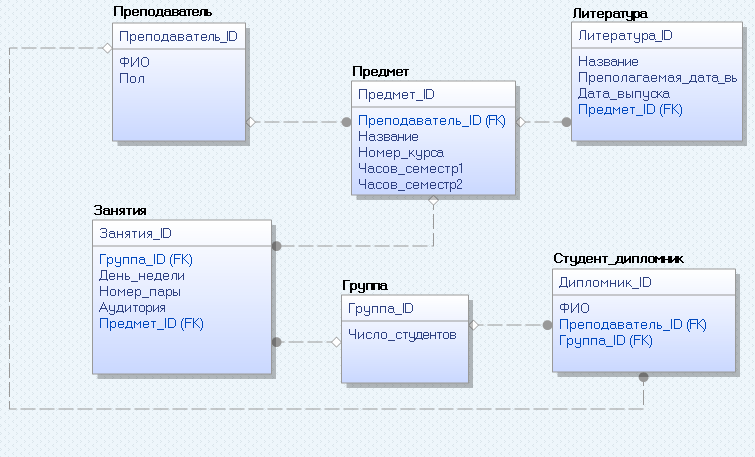


Рис. 1.1. Реляционная модель

# Глава 2. Реализация БД

## СУБД Microsoft SQL Server

Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с небольшими и средними по размеру базами данных до крупных баз данных масштаба предприятия.

## Описание физической модели БД

Описание физической модели сгенерировано с помощью средств ERwin Data Modeler. Приведем создание таблиц(остальная часть реализации достаточно объемна).

**CREATE TABLE** [Группа] (

[Группа\_ID] [Число\_студентов]

integer **NOT NULL** ,

integer **NOT NULL**

**CONSTRAINT** [positive\_2089158866]

**CHECK** ( [Число\_студентов] > 0 )

)

**go**

**CREATE TABLE** [Занятия] (

[Занятия\_ID] integer **NOT NULL** ,

[Группа\_ID] integer **NULL** ,

[День\_недели] char(18) **NOT NULL CONSTRAINT** [week\_days\_717133032]

**CHECK** ( [День\_недели]='пн' **OR** [День\_недели]='вт' **OR** [День\_недели]='ср' **OR** [День\_недели]='чт' **OR** [День\_недели]='пт' **OR** [День\_недели]='сб' ),

[Номер\_пары] integer **NOT NULL CONSTRAINT** [1\_7\_81269735]

**CHECK** ( Номер\_пары **BETWEEN** 1 **AND** 7 ), [Аудитория] char(18) **NOT NULL** ,

[Предмет\_ID] integer **NULL**

)

**go**

**CREATE TABLE** [Литература] (

[Литература\_ID] [Название]

integer **NOT NULL** ,

char(100) **NOT NULL** ,

[Преполагаемая\_дата\_выпуска] datetime **NOT NULL** , [Дата\_выпуска] datetime **NULL** , [Предмет\_ID] integer **NULL**

)

**go**

**CREATE TABLE** [Предмет] (

[Предмет\_ID] [Преподаватель\_ID] [Название] [Номер\_курса]

integer

integer

**NOT NULL** ,

**NULL** ,

char(100) **NOT NULL** ,

integer **NOT NULL**

**CONSTRAINT** [1\_5\_1897663438]

**CHECK** ( Номер\_курса **BETWEEN** 1 **AND** 5 ), [Часов\_семестр1] integer **NOT NULL CONSTRAINT** [sem\_hours\_1969219323]

**CHECK** ( Часов\_семестр1 **BETWEEN** 0 **AND** 120 ), [Часов\_семестр2] integer **NOT NULL**

**CONSTRAINT** [sem\_hours\_1969219579]

**CHECK** ( Часов\_семестр2 **BETWEEN** 0 **AND** 120 )

)

**go**

**CREATE TABLE** [Преподаватель] (

[Преподаватель\_ID] integer **NOT NULL** , [ФИО] char(150) **NOT NULL** ,

[Пол] char(1) **NOT NULL CONSTRAINT** [FemaleOrMale]

**CHECK** ( [Пол]='м' **OR** [Пол]='ж' )

)

**go**

**CREATE TABLE** [Студент\_дипломник] (

[Дипломник\_ID] integer **NOT NULL** , [ФИО] char(100) **NOT NULL** ,

[Преподаватель\_ID] integer **NULL** , [Группа\_ID] integer **NULL**

)

**go**

Запросы, реализующие функционал БД, представлены в виде хранимых процедур. Все процедуры описаны в Приложении 2.

Пример процедуры поиска расписания для преподавателя:

**create procedure** teacherTimetable(@teacher varchar(100))

**as begin**

**select** t3.Преподаватель\_ID, t3.ФИО, t2.Название, t1.День\_недели, t1.Номер\_пары, t1.Группа\_ID, t1.Аудитория

**from** Занятия t1 **join** Предмет t2

**on** t1.Предмет\_ID = t2.Предмет\_ID

**join**

Преподаватель t3

**on** t2.Преподаватель\_ID = t3.Преподаватель\_ID **where** t3.ФИО **like** '%' + @teacher + '%' **order by** t1.День\_недели, t1.Номер\_пары

**end**

# Глава 3. Реализация базы данных

## Описание разработанного приложения

Главное форма приложения содержит меню, из которого можно перейти в пункты Таблицы и Запросы.

Пункт Таблицы обладает подпунктами, позволяющими перейти на формы с соответствующими таблицами. Каждая форма обладает

DataGridView, которая подключена к соответствующей таблице, и кнопкой применения изменений к таблице базы данных, внесенных в DataGridView. Пункт Запросы обладает подпунктами, позволяющими перейти на формы с соответствующими запросами. Каждая форма имеет поля ввода параметров процедуры, соответствующей запросу, если они есть. Также имеется кнопка вызова процедуры, вывод происходит в DataGridView без

возможностей редактирования.

Примеры экранных форм приведены в Приложении 3. Листинг программы, реализующей приложение для работы с БД приведен в Приложении 4.

# Заключение

В ходе выполнения курсового проекта:

* был проведен анализ предметной области для БД и выделены основные информационные объекты;
* разработана ER- модель БД, в которой выделены 6 сущностей и определены связи между ними;
* разработана реляционная модель БД, включающая 6 таблиц. Показано, что модель находится в требуемой третьей нормальной форме;
* разработана база данных в среде MS SQL Server и 6 хранимых процедур, реализующих заданные функции работы с БД;
* разработано приложение с БД в на языке С# с технологией ADO.N