**

**ИНСТИТУТ ТЕХНИКИ И ЦИФРОВЫХ ТЕХНОЛОГИЙ**

**ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ**

**Курсовой проект**

по дисциплине «Базы данных»

на тему ***«Разработка базы данных для поддержки процессов работы электронного журнала»***

**Вариант №3**

Проект выполнил студент (ФИО) Бондаренко Мария Алексеевна

группы ИО-17

(подпись)

Проект защищен с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Защиту принял Ф.И.О. преподавателя

(подпись)

**Королев 2019**

Оглавление

[Введение 3](#_Toc28116651)

[1. Постановка задачи 3](#_Toc28116652)

[2. Анализ предметной области 3](#_Toc28116653)

[3. Инфологическая модель БД 4](#_Toc28116654)

[4. Логическая модель БД 7](#_Toc28116655)

[5. Описание программных средств реализации БД 9](#_Toc28116656)

[5.1. СУБД Microsoft SQL Server 9](#_Toc28116657)

[5.2. Описание физической модели БД 11](#_Toc28116658)

[5.3. Язык Высокого Уровня С#. Технология ADO .NET 12](#_Toc28116659)

[6. Описание разработанного приложения 13](#_Toc28116660)

[6.1. Таблицы 13](#_Toc28116661)

[6.2. Формы 13](#_Toc28116662)

[6.3. Запросы 15](#_Toc28116663)

[Заключение 17](#_Toc28116664)

[Список используемой литературы 18](#_Toc28116665)

# Введение

Цель курсового проекта заключается в создании базы данных (БД) для выбранной предметной области, а также приложения для работы с таблицами.

Для решения поставленной задачи выбрана система управления реляционными базами данных (СУБД) Microsoft SQL Server Management Studio 18 и язык высокого уровня (ЯВУ) C#, на котором реализован программный продукт.

В результате курсового проекта создан проект, реализующий поставленную задачу, и отчет, подтверждающий обоснованность принятых при проектировании решений.

# Постановка задачи

Разработка БД для поддержки процессов работы электронного журнала. Задачи, решаемые в ходе выделенного процесса:

* учет дисциплин, закрепленных за каждой группой студентов;
* учет преподавателей, заполняющих журнал;
* учет посещения занятий;
* контроль выполнения различных видов заданий для каждой дисциплины (лабораторных работ, практических заданий, ТР, КП);
* оценка текущей успеваемости
* расчет рейтинга студента по каждой дисциплине
* составление списка неуспевающих студентов.

# Анализ предметной области

Исходя из анализа процесса в заданной предметной области и выделенных задач, решаемых в ней, можно выделить следующие информационные объекты: студент, группа студентов, преподаватель, дисциплина, посещение, контроль выполнения различных видов заданий, успеваемость. Для построения моделей БД используется Microsoft Visio.

# Инфологическая модель БД

Выделенные в процессе анализа предметной области информационные объекты отображаются в сущности ER-модели. Графическая ER-модель представлена на рис. 1. Описание сущностей БД приведено в табл. 1.

Таблица 1. Описание ER-модели

|  |  |  |
| --- | --- | --- |
| **Описание сущностей** | | |
| **№** | **Имя** | **Описание** |
| 1 | Student | Студент |
| 2 | Grp | Группа студентов |
| 3 | Tchr | Преподаватель |
| 4 | Disc | Дисциплина |
| 5 | Grp\_disc | Список предметов каждой группы |
| 6 | Stud\_jour | Студент в журнале, учёт его оценок и посещаемости |
| 7 | Journal | Журнал, расписание занятий |
| 8 | Lesson | Вид занятия (практика, лекция и пр.) |
| 9 | Tchr\_disc | Закрепление дисциплин за преподавателями |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибуты сущности «Student»** | | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Номер студенческого билета | Id\_Stud | integer | Yes | No |
| Название группы | Name\_Grp | varchar(10) | No | Yes |
| ФИО студента | Name\_Stud | varchar(20) | No | No |
| Контактные данные | Cont | varchar(20) | No | No |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибуты сущности «Grp»** | | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Название группы | Name\_Grp | varchar(10) | Yes | No |
| Полное название группы | FullName | varchar(50) | No | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибуты сущности «Tchr»** | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Идентификационный номер преподавателя | Id\_Tch | integer | Yes | No |
| ФИО преподавателя | Name\_Tch | varchar(20) | No | No |
| Контактные данные | Cont\_Tch | varchar(20) | No | No |
| Должность | Dol | varchar(20) | No | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибуты сущности «Disc»** | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Идентификационный номер дисциплины | Id\_Disc | integer | Yes | No |
| Название дисциплины | Name\_Disc | varchar(30) | No | No |
| Кол-во занятий | Num | integer | No | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибуты сущности «Grp\_disc»** | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Идентификационный номер | Id\_GP | integer | Yes | No |
| Название группы | Name\_Grp | varchar(10) | No | Yes |
| Идентификационный номер дисциплины | Id\_Disc | integer | No | Yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибуты сущности «Stud\_jour»** | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Идентификационный номер | Id\_SJ | integer | Yes | No |
| Идентификационный номер в журнале | Id\_Jour | integer | No | Yes |
| Номер студенческого билета | Id\_Stud | integer | No | Yes |
| Оценка (от 0 до 5) | Grade | integer | No | No |
| Посещение (1 – посещал занятие, 0 – не посещал) | Visit | integer | No | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибуты сущности «Journal»** | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Идентификационный номер в журнале | Id\_Jour | integer | Yes | No |
| Идентификационный номер дисциплины | Id\_Disc | integer | No | Yes |
| Идентификационный номер занятия | Id\_less | integer | No | Yes |
| Дата занятия | Dat | datetime | No | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибуты сущности «Lesson»** | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Идентификационный номер занятия | Id\_less | Id\_less | Yes | No |
| Название занятия | Name\_Less | varchar(20) | No | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибуты сущности «Tchr\_disc»** | | | | |
| **Описание** | **Имя** | **Тип данных** | **Is PK** | **Is FK** |
| Идентификационный номер | Id\_TD | integer | Yes | No |
| Идентификационный номер дисциплины | Id\_Disc | integer | No | Yes |
| Идентификационный номер преподавателя | Id\_Tch | integer | No | Yes |

# Логическая модель БД

На основе построенной ER-модели для разработки БД создается реляционная модель (РМД). Для её построения необходимо выполнить несколько этапов:

* преобразовать сущности ER-модели в отношения (таблицы) РМД;
* привести отношения к 3НФ;
* определить правила целостности для отношений РМД.

В результате выполнения указанных этапов была получена РМД БД «Электронный журнал». Её графическая модель представлена на рис.1. Описание отношений, их атрибутов, ключей и статических правил целостности приведены в табл. 1.

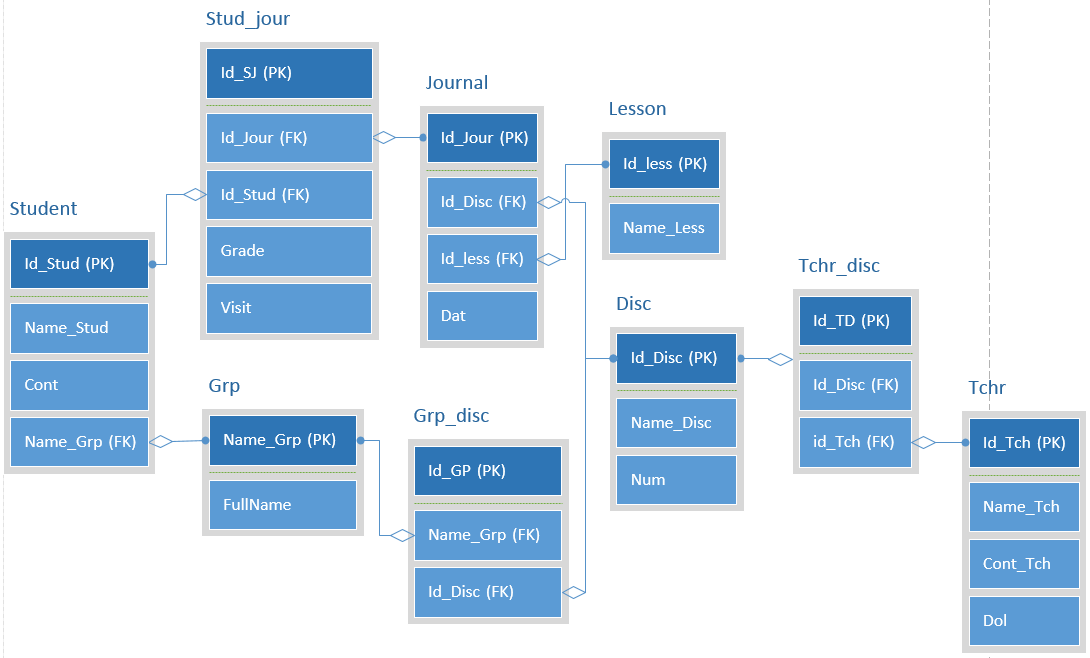


Рис. 1 – Реляционная модель БД

Для обоснования нахождения БД в 3НФ необходимо провести анализ функциональных зависимостей (ФЗ) спроектированных отношений. Ниже приводятся ФЗ для каждого отношения.

Схема отношения – **Student** (Id\_Stud, Name\_Stud, Cont, Name\_Grp), ключ отношения – Id\_Stud.

Функциональные зависимости:

Id\_Stud →Name\_Stud,

Id\_Stud →Cont,

Id\_Stud →Name\_Grp

**Grp** (Name\_Grp, FullName), ключ – Name\_Grp.

Функциональные зависимости:

Name\_Grp → FullName

**Grp\_disc** (Id\_GP, Name\_Grp, Id\_Disc), ключ - Id\_GP

Функциональные зависимости:

Id\_GP → Name\_Grp,

Id\_GP → Id\_Disc

**Disc** (Id\_Disc, Name\_Disc, Num), ключ – Id\_Disc

Функциональные зависимости:

Id\_Disc → Name\_Disc,

Id\_Disc →Num

**Tchr\_disc** (Id\_TD, Id\_Disc, Id\_Tch), ключ – Id\_TD

Функциональные зависимости:

Id\_TD → Id\_Disc,

Id\_TD → Id\_Tch

**Tchr** (Id\_Tch, Name\_Tch, Cont\_Tch, Dol), ключ - Id\_Tch

Функциональные зависимости:

Id\_Tch → Name\_Tch,

Id\_Tch → Cont\_Tch,

Id\_Tch → Dol

**Stud\_jour** (Id\_SJ, Id\_Jour, Id\_Stud, Grade, Visit), ключ - Id\_SJ

Функциональные зависимости:

Id\_SJ → Id\_Jour,

Id\_SJ → Id\_Stud,

Id\_SJ → Grade,

Id\_SJ → Visit

**Journal** (Id\_Jour, Id\_Disc, Id\_less, Dat), ключ - Id\_Jour

Функциональные зависимости:

Id\_Jour → Id\_Disc,

Id\_Jour → Id\_less,

Id\_Jour → Dat

**Lesson** (Id\_less, Name\_Less), ключ - Id\_less

Функциональные зависимости:

Id\_less → Name\_Less

Других ФЗ для отношений не выявлено. Обоснование нахождения БД в 3НФ предполагает проверку ограничений, накладываемых в 3НФ.

1. Все таблицы БД находятся в 1НФ, их поля содержат атомарные значения.
2. Полная ФЗ неключевых атрибутов от ключа для отношений Student, Grp, Grp\_disc, Disc, Tchr\_disc, Tchr, Stud\_jour, Journal, Lesson обеспечивается тем, что ключи в этих отношениях простые.
3. В разработанных отношениях отсутствуют транзитивные зависимости.

Следовательно, БД находится в 3НФ.

# Описание программных средств реализации 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; конкурирует с другими СУБД в этом сегменте рынка.

SQL является общепринятым интерфейсом к базам данных. «Все промышленные базы — Oracle, Microsoft SQL Server, PostgreSQL, MySQL — работают на SQL.

***Функциональность***

**Microsoft SQL Server** в качестве языка запросов использует версию SQL, получившую название Transact-SQL (сокращённо T-SQL), являющуюся реализацией SQL-92 (стандарт ISO для SQL) с множественными расширениями. T-SQL позволяет использовать дополнительный синтаксис для хранимых процедур и обеспечивает поддержку транзакций (взаимодействие базы данных с управляющим приложением). Microsoft SQL Server и Sybase ASE для взаимодействия с сетью используют протокол уровня приложения под названием Tabular Data Stream (TDS, протокол передачи табличных данных). Протокол TDS также был реализован в проекте FreeTDS с целью обеспечить различным приложениям возможность взаимодействия с базами данных Microsoft SQL Server и Sybase.

Microsoft SQL Server также поддерживает Open Database Connectivity (ODBC) — интерфейс взаимодействия приложений с СУБД. Версия SQL Server 2005 обеспечивает возможность подключения пользователей через веб-сервисы, использующие протокол SOAP. Это позволяет клиентским программам, не предназначенным для Windows, кроссплатформенно соединяться с SQL Server. Microsoft также выпустила сертифицированный драйвер JDBC, позволяющий приложениям под управлением Java (таким как BEA и IBM WebSphere) соединяться с Microsoft SQL Server 2000 и 2005.

Также в случае отказа или сбоя на одном из серверов кластера доступен автоматический перенос нагрузки на другой сервер.

## Описание физической модели БД

Операторы описания модели БД MS SQL Server

CREATE TABLE Student(

Id\_Stud int NOT NULL PRIMARY KEY,

Name\_Grp varchar (10) FOREIGN KEY REFERENCES Grp(Name\_Grp) ON DELETE CASCADE,

Name\_Stud varchar (20),

Cont varchar (20),

);

CREATE TABLE Tchr(

Id\_Tch int NOT NULL PRIMARY KEY,

Name\_Tch varchar (20),

Cont\_Tch varchar (20),

Dol varchar (20)

);

CREATE TABLE Grp(

Name\_Grp varchar (10) NOT NULL PRIMARY KEY,

FullName varchar (50)

);

CREATE TABLE Grp\_disc(

Id\_GD int NOT NULL PRIMARY KEY,

Name\_Grp varchar (10) FOREIGN KEY REFERENCES Grp(Name\_Grp) ON DELETE CASCADE,

Id\_Disc int FOREIGN KEY REFERENCES Disc(Id\_Disc) ON DELETE CASCADE

);

CREATE TABLE Disc(

Id\_Disc int NOT NULL PRIMARY KEY,

Name\_Disc varchar (30),

Num int

);

CREATE TABLE Tchr\_disc(

Id\_TD int NOT NULL PRIMARY KEY,

Id\_Disc int FOREIGN KEY REFERENCES Disc(Id\_Disc) ON DELETE CASCADE,

Id\_Tch int FOREIGN KEY REFERENCES Tchr(Id\_Tch) ON DELETE CASCADE

);

CREATE TABLE Lesson(

Id\_less int NOT NULL PRIMARY KEY,

Name\_Less varchar (20)

);

CREATE TABLE Journal(

Id\_Jour int NOT NULL PRIMARY KEY,

Id\_Disc int FOREIGN KEY REFERENCES Disc(Id\_Disc) ON DELETE CASCADE,

Id\_less int FOREIGN KEY REFERENCES Lesson(Id\_less) ON DELETE CASCADE,

Dat datetime

);

CREATE TABLE Stud\_jour(

Id\_SJ int NOT NULL PRIMARY KEY,

Id\_Jour int FOREIGN KEY REFERENCES Journal(Id\_Jour) ON DELETE CASCADE,

Id\_Stud int FOREIGN KEY REFERENCES Student(Id\_Stud) ON DELETE CASCADE,

Grade int, Visit int );

## Язык Высокого Уровня С#. Технология ADO .NET

**ADO.NET** (ActiveX Data Object для .NET) — технология, предоставляющая доступ и управление данными, хранящимся в базе данных или других источниках (Microsoft SQL Server, Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft Exchange, Oracle, OLE DB, ODBC, XML, текстовые файлы), основанных на платформе .NET Framework и входящая в состав .NET Framework 2.0, представляет собой набор библиотек. В отличие от технологии ADO, которая была в основном предназначена для тесно связанных клиент-серверных систем, ADO.NET больше нацелена на автономную работу с помощью объектов DataSet. Объекты DataSet представляют локальные копии взаимосвязанных таблиц данных, каждая из которых содержит набор строк и столбцов. Объекты DataSet позволяют вызывающей сборке (наподобие веб-страницы или программы, выполняющейся на настольном компьютере) работать с содержимым DataSet, изменять его, не требуя подключения к источнику данных, и отправлять обратно блоки измененных данных для обработки с помощью соответствующего адаптера данных.

**Поставщики данных**

Технология ADO.NET построена так, чтобы изолировать программиста от изучения структур баз данных разных производителей, представляя поставщиков баз данных (data provider), которые инкапсулируют механизм работы с конкретной СУБД, что позволяет создавать адаптеры для любой СУБД и полностью использовать её особенности. Сделана такая абстракция для того, чтобы использовать одинаковые типы данных для работы с различными источниками данных, иметь общий подход (универсализацию) для работы с базами данных разных производителей, чтобы технология ADO.NET поддерживалась CLR.

# Описание разработанного приложения

## Таблицы

Описание таблиц БД, реализованных в СУБД MS SQL Server приведены на рис. 2.

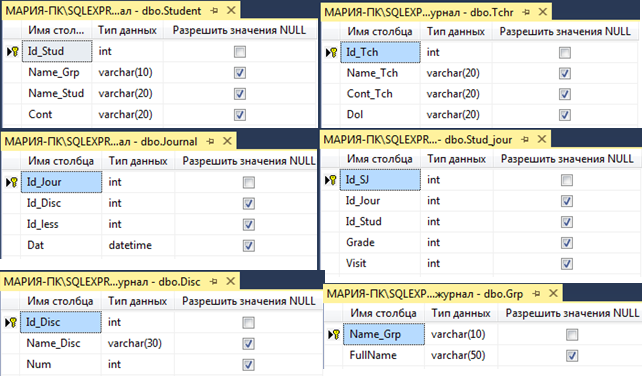


Рис. 2 – Структура таблиц БД «Электронный журнал»

## Формы

Пример реализации интерфейсных форм представлен на рис. 3 – 6. Главное окно с вкладками «Таблицы» и «Запросы». Во вкладке «Таблицы» справа расположены области с кнопками «Просмотр» и «Команды». В области «Просмотр» при нажатии на одну из кнопок будет выведена таблица, название которой соответствует названию нажатой кнопки.

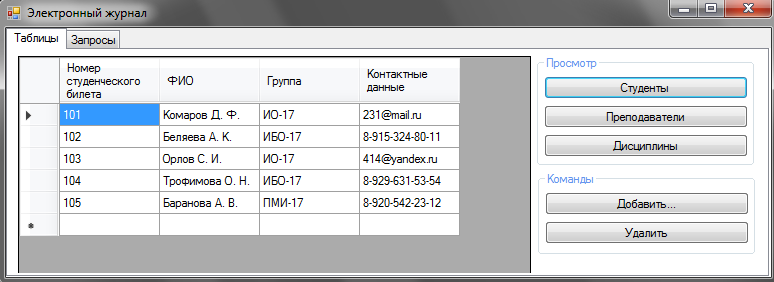


Рис. 3 – Главная форма приложения с БД «Электронный журнал»

При нажатии кнопки «Добавить…» откроется новое окно «Добавление данных в таблицу…». Добавление данных будет происходить в той таблице, что была открыта для просмотра.

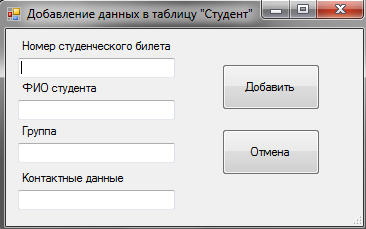


Рис. 4 – Форма добавления данных в таблицу «Студент»

При нажатии кнопки «Удалить» откроется окно с подтверждением данной операции. Удалена будет выделенная пользователем строка в текущей таблице.

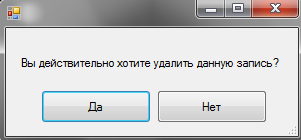


Рис. 5 – Окно подтверждение удаления выбранной записи таблицы

Во вкладке «Запросы» сверху расположены две области «Запросы» и «Фильтр». Первый фильтр соответствует запросу «Контроль выполнения различных видов заданий» и позволяет выбирать необходимую дисциплину и вид задания. Второй фильтр соответствует запросу «Рейтинг студента по каждой дисциплине» и позволяет выбирать необходимую дисциплину и ФИО студента.

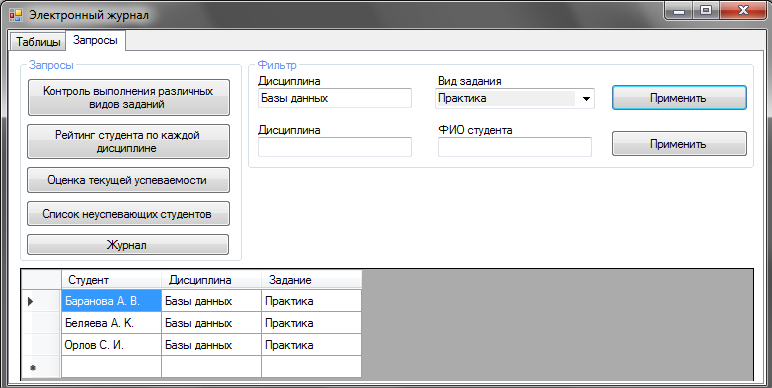


Рис. 6 – Вкладка «Запросы» в главной форме

## Запросы

Запросы для загрузки данных в таблицу

"SELECT Id\_Stud as 'Номер студенческого билета', Name\_Stud as 'ФИО', Name\_Grp as 'Группа', Cont as 'Контактные данные' FROM Student";

"SELECT id\_Tch as 'Идентификационный номер', Name\_Tch as 'ФИО', Dol as 'Должность', Cont\_Tch as 'Контактные данные' FROM Tchr";

SELECT Id\_Disc as 'Идентификационный номер', Name\_disc as 'Название', Num as 'Кол-во занятий' FROM Disc";

" SELECT Id\_SJ as 'ID записи', Journal.Id\_Jour as 'День №', Student.Name\_Stud as 'Студент', "+

" Disc.Name\_Disc as 'Дисциплина', Lesson.Name\_Less as 'Занятие' , Grade as 'Оценка', Visit as 'Посещение' FROM Stud\_jour" +

" INNER JOIN Journal ON(Journal.Id\_Jour= Stud\_jour.Id\_jour)" +

" INNER JOIN Student ON(Student.Id\_Stud= Stud\_jour.Id\_Stud)" +

" INNER JOIN Disc ON(Disc.Id\_Disc= Journal.Id\_Disc)" +

" INNER JOIN Lesson ON(Lesson.Id\_less= Journal.Id\_less);";

Запросы для добавления данных в таблицу

"INSERT INTO Student (Id\_Stud, Name\_Stud, Name\_Grp, Cont) " +

"VALUES (" + id + ", '" + name + "', '" + nameGr + "', " + cont + ")";

"INSERT INTO Tchr (Id\_Tch, Name\_Tch, Cont\_Tch, Dol) " +

"VALUES (" + id + ", '" + name + "', '" + cont + "', '" + dol + "')";

"INSERT INTO Disc (Id\_Disc, Name\_Disc, Num) " +

"VALUES (" + id + ", '" + name + "'," + num + ")";

Запросы для удаления данных в таблицу

"DELETE FROM Student WHERE Student.Id\_Stud = " + ID;

"DELETE FROM Tchr WHERE Tchr.Id\_Tch = " + ID;

"DELETE FROM Disc WHERE Disc.Id\_Disc = " + ID;

"DELETE FROM Journal WHERE Journal.Id\_Jour = " + ID;

Контроль выполнения различных видов заданий для каждой дисциплины (лабораторных работ, практических заданий, ТР, КП)

"SELECT Name\_Stud as 'Студент', Name\_Disc as 'Дисциплина', Name\_Less as 'Задание' FROM Disc"

+ " INNER JOIN Journal ON(Disc.Id\_Disc= Journal.Id\_Disc)"

+ " INNER JOIN Stud\_jour ON(Journal.Id\_Jour= Stud\_jour.Id\_Jour)"

+ " INNER JOIN Lesson ON(Lesson.Id\_less= Journal.Id\_less)"

+ " INNER JOIN Student ON(Student.Id\_Stud= Stud\_Jour.Id\_Stud)"

+ " WHERE Grade<> 0 AND Journal.Id\_less > 601"

+ " GROUP BY Name\_Disc, Name\_Stud, Name\_Less, Grade, Journal.Id\_less;";

Оценка текущей успеваемости

"SELECT Name\_Stud as 'Студент', SUM(Grade)+SUM(Visit) as 'Баллы' FROM Student" +

" INNER JOIN Stud\_jour ON(Student.Id\_Stud= Stud\_jour.Id\_Stud)" +

" GROUP BY Name\_Stud ORDER BY 'Баллы' DESC;";

Расчет рейтинга студента по каждой дисциплине

"SELECT Name\_Stud as 'Студент', Name\_Disc as 'Дисциплина', " +

"SUM(Grade)+SUM(Visit) as 'Баллы' FROM Disc " +

"INNER JOIN Journal ON(Disc.Id\_Disc= Journal.Id\_Disc) " +

"INNER JOIN Stud\_jour ON(Journal.Id\_Jour= Stud\_jour.Id\_Jour) " +

"INNER JOIN Lesson ON(Lesson.Id\_less= Journal.Id\_less) " +

"INNER JOIN Student ON(Student.Id\_Stud= Stud\_Jour.Id\_Stud) " +

"GROUP BY Name\_Disc, Name\_Stud " +

"ORDER BY Name\_Stud, 'Баллы' DESC;";

Составление списка неуспевающих студентов

"SELECT TOP 2 Name\_Stud as 'Студент', SUM(Grade)+SUM(Visit) as 'Баллы' FROM Student" +

" INNER JOIN Stud\_jour ON(Student.Id\_Stud= Stud\_jour.Id\_Stud) GROUP BY Name\_Stud ORDER BY 'Баллы' ASC; ";

# Заключение

В ходе выполнения курсового проекта:

* был проведен анализ предметной области для БД и выделены основные информационные объекты;
* разработана ER-модель БД, в которой выделено 9 сущностей и определены связи между ними;
* разработана реляционная модель БД, включающая 9 таблиц. Модель БД доведена до требуемой третьей нормальной формы;
* разработана БД в среде MS SQL Server и 7 запросов, реализующих заданные функции работы с БД;
* разработано приложение с БД на языке С# с технологией ADO.Net

# Список используемой литературы

1. Сидорова, Н.П. Основы T-SQL. Практикум по курсу "Базы данных" [Текст] : учебное пособие / Н.П. Сидорова, Г.Н. Исаева. - Королев М.О. : МГОТУ, 2017. - 68 с. - ISBN 978-5-91730-784-8.
2. Дьюсон Р. SQL Server 2008 для начинающих разработчиков. Спб.: БХВПетербург, 2009. – 704 с.
3. Пирогов, В.Ю. Информационные системы и базы данных: Организация и проектирование. СПб. : БХВ-Петербург, 2009 – 528 с.
4. Гущин А.Н. Базы данных М.: Директ-медиа, -2015, - 266с. http://biblioclub.ru/index.php?page=book\_red&id=278093&sr=1
5. Шустова Л.И., Тараканов О.В Базы данных: Учебник - М.: НИЦ ИНФРА-М, 2016. - 304 с- (Высшее образование: Бакалавриат) ISBN 978-5-16-010485-0, http://znanium.com/bookread2.php?book=491069
6. Мартишин С.А., Симонов В.Л., Храпченко М.В Базы данных. Практическое применение СУБД SQL и NoSOL-типа для применения проектирования информационных систем: Учебное пособие. - М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2017. http://znanium.com/bookread2.php?book=556449