

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Дмитриев Николай Николаевич  
Должность: Ректор  
Дата подписания: 24.02.2025 04:11  
Уникальный программный ключ:  
f7c6227919e4cdbfb4d7b682991f8553b37cafb

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ им. А.А.  
ЕЖЕВСКОГО

Кафедра «Информатика и математическое моделирование»

# ***РАЗРАБОТКА БАЗ ДАННЫХ В MICROSOFT SQL SERVER***

Учебно-методические указания

к выполнению практических работ

ПМ.07 Соадминистрирование и автоматизация баз данных и серверов

для студентов специальности:

Специальность: 09.02.07 Информационные системы и программирование

Молодежный, 2022

Печатается по решению методической комиссии института экономики, управления и прикладной информатики Иркутского государственного аграрного университета им. А.А. Ежевского.

Протокол №3 от 14 января 2022 г.

**Рецензенты:** к.т.н., доцент, директор института экономики, управления и прикладной информатики Федурин Н.И.; доцент кафедры информатики и математического моделирования Беляков А.Ю.

Данные методические указания разработаны для поддержки практических занятий и самостоятельной работы по курсу «Сoadминистрирование и автоматизация баз данных и серверов».

В качестве среды программирования используется Microsoft SQL Server 2014 (или более поздняя версия).

© Бендик Н.В. 2022  
© Иркутский ГАУ, 2022

## Введение

В настоящее время информационные системы, применяющие базы данных, представляют собой одну из важнейших областей современных компьютерных технологий. При построении корпоративных систем обработки данных формируется единое информационное пространство, работа в котором носит распределенный характер. Распределенная обработка данных реализуется в компьютерных сетях и требует определенной дисциплины взаимодействия. Общепринятым стандартом такого взаимодействия стала технология клиент-сервер, когда часть функций прикладной программы реализована на программе-клиенте, другая – на программе-сервере.

Серверы баз данных являются наиболее эффективным инструментом для создания приложений, оперирующими большими объемами информации и являются важнейшим звеном в их построении по схеме клиент-сервер. При этом эффективно реализуется интегрированность базы данных, то есть возможность одновременного доступа к данным из нескольких приложений. Создание клиент-серверного приложения, работающего с базой данных, требует прохождения следующих этапов:

- 1 – разработка структуры реляционной базы данных;
- 2 – администрирование базы данных на стороне сервера;
- 3 – программирование на стороне сервера;
- 4 – программирование на стороне клиента.

В данном методическом руководстве рассматриваются все этапы построения клиент-серверного приложения на примере создания системы делопроизводства деканата вуза. В настоящий момент, одной из самых популярных серверных платформ, является Microsoft SQL Server, на котором и построена рассматриваемая база данных. Клиентское приложение строится на

основе СУБД Microsoft Visual Foxpro. Foxpro является специализированным языком программирования, который ориентирован на работу с базами данных. Эта программная среда позволяет создавать эффективные приложения, работающие с локальными, сетевыми и удаленными базами данных на разных серверных платформах, в том числе и на Microsoft SQL Server.

В предлагаемой работе ввод данных и их программирование осуществляется и на стороне клиента, и на стороне сервера. Используются все средства, входящие в состав Visual Foxpro для формирования клиент-серверного приложения и взаимодействия с сервером баз данных на основе Microsoft SQL Server. Взаимодействие клиентской и серверной части разрабатываемой базы данных построено таким образом, чтобы продемонстрировать основные возможности Microsoft SQL Server по управлению входящими потоками информации:

- с помощью триггеров, программируемых на стороне сервера, повышается скорость обработки входящих данных и снижается трафик в сети организации;
- разработка транзакций на сервере позволяет производить откат некорректно выполненных операций;
- с помощью средств мониторинга и сопровождения SQL Server решается ряд задач по администрированию сервера.

Таким образом целью данной работы является рассмотрение основных возможностей Microsoft SQL Server по управлению базой данных и решение основных задач по администрированию сервера, построение клиентских приложений для серверных платформ и организация взаимодействия сервера с клиентским программным обеспечением.

## Часть I. Разработка базы данных на Microsoft SQL Server.

Разработка клиент-серверной информационной системы начинается с разработки базы данных на стороне сервера и настройки серверной платформы. Здесь можно выделить следующие задачи:

- 1 – создание базы данных и установка ее свойств;
- 2 - разработка таблиц;
- 3 – установление отношений между таблицами и обеспечение целостности данных;
- 4 – программирование на стороне сервера, написание триггеров и транзакций;
- 5 - ввод первоначальных данных.

### 1.1.Создание базы данных и установка ее свойств.

Создайте новую базу данных.

Откройте Enterprise Manager, найдите в консоли папку Databases на сервере, который вы используете. SQL Server отобразит список баз данных (рис.1.1).

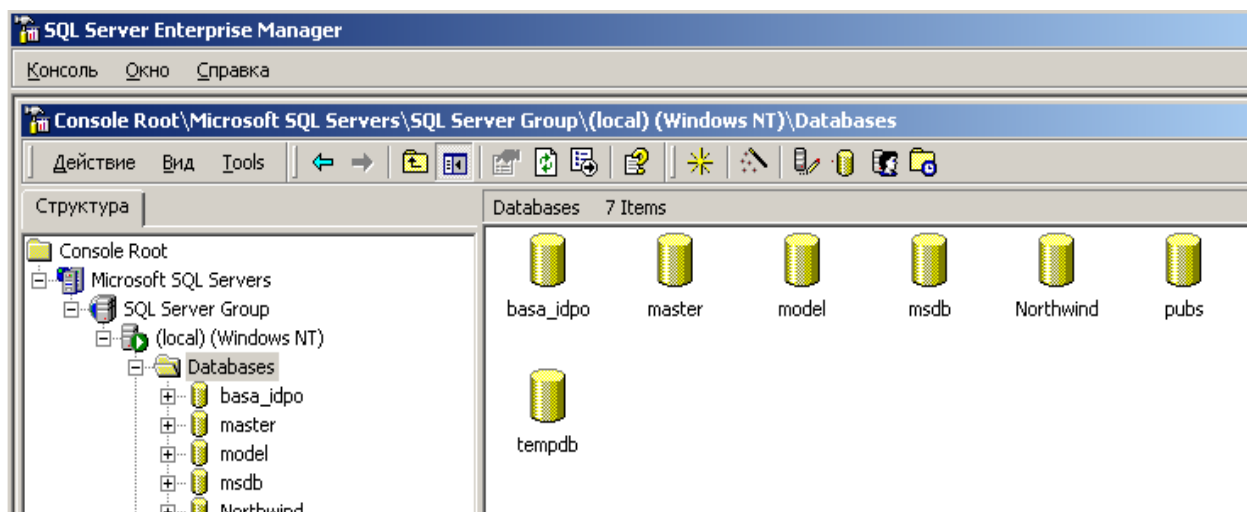


Рис.1.1 Создание базы данных

Выберите инструмент «New Database» и введите имя создаваемой базы данных.

Размещение файлов базы данных.

В процессе эксплуатации базы данных неоднократно возникают задачи переноса базы данных с одного носителя на другой и управления производительностью работы сервера. При создании новой базы данных формируются два файла ее сопровождения:

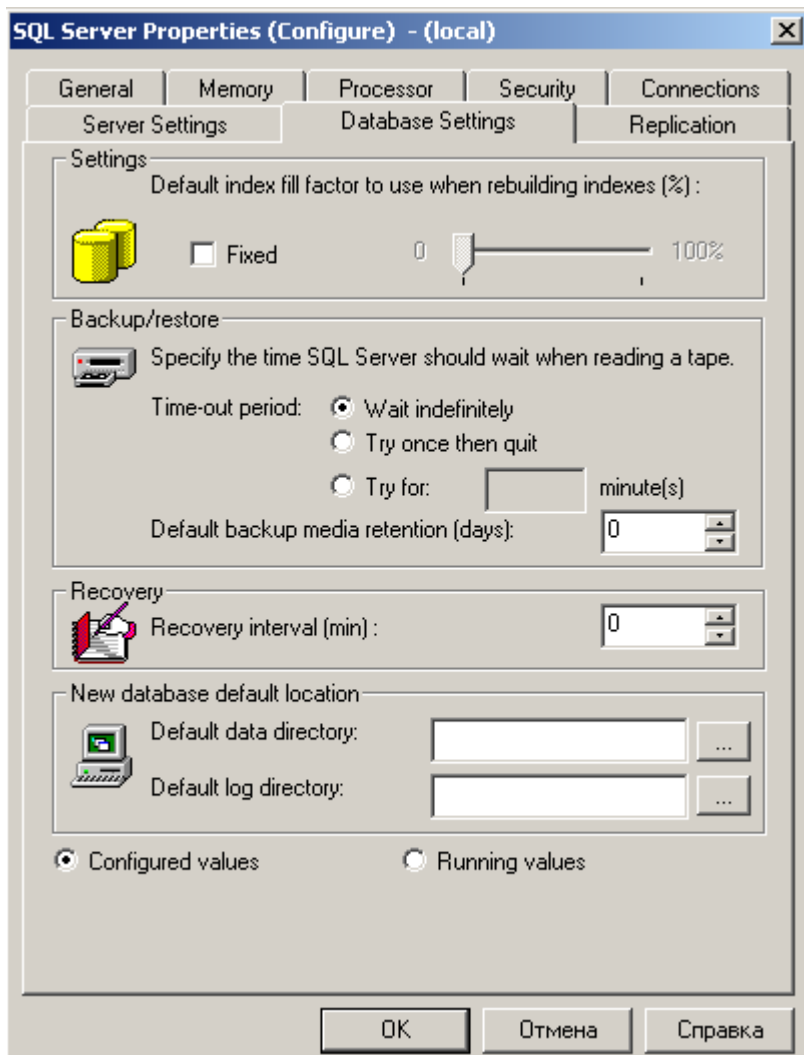
\*.MDF (Master Data File)– файл данных, предназначен для хранения информации, находящейся в таблицах базы данных. Кроме того, в этом файле так же размещены процедуры, ограничения, триггеры, индексы и другая информация;

\*.LDF – файл журнала транзакций, хранит информацию о ходе выполнения транзакций. В нем размещается информация о состоянии данных перед началом транзакции, о выполняемых изменениях, заблокированных ресурсах и другая сопутствующая информация.

Любая база данных должна содержать как минимум один файл данных и один файл журнала транзакций. При необходимости администратор может добавлять в БД новые файлы данных или файлы журнала транзакций. Если компьютер, на котором установлен SQL Server имеет несколько физических дисков, то для повышения производительности Microsoft настоятельно рекомендует для каждой базы данных создать как минимум один файл на каждом физическом диске. Кроме того, по возможности следует располагать файлы данных и журнала транзакций на отдельных физических дисках. Это повышает производительность работы всего сервера баз данных.

При первой установке SQL Server для новых баз данных по умолчанию принимается место размещения: \диск установки SQL Server\Program Files\Microsoft SQL Server\MSSQL\data. Эта настройка по умолчанию предоставляется мастеру создания базы данных Create Database Wizard. Чтобы изменить эту используемую по умолчанию установку, вы можете задать новое место размещения на вкладке Database Settings (Параметры базы данных) в диалоговом окне SQL Server Properties (Свойства SQL Server). Для этого:

1. Щелкните правой кнопкой мыши на сервере в дереве консоли Console Tree, выберите Properties (Свойства), а затем откройте вкладку Database Settings (Параметры базы данных).



2. Перейдите к разделу Default data directoty и щелкните на кнопке Browse (Обзор), чтобы изменить местоположение файла базы данных. Мастер отобразит диалоговое окно, запрашивающее новое место размещения. Укажите нужную вам папку для размещения файла базы данных.

3. Перейдите к разделу Default log directoty и щелкните на кнопке Browse (Обзор), чтобы изменить местоположение файла журнала транзакций. Мастер отобразит диалоговое окно, запрашивающее новое место размещения. Укажите нужную вам папку для размещения файла базы данных.



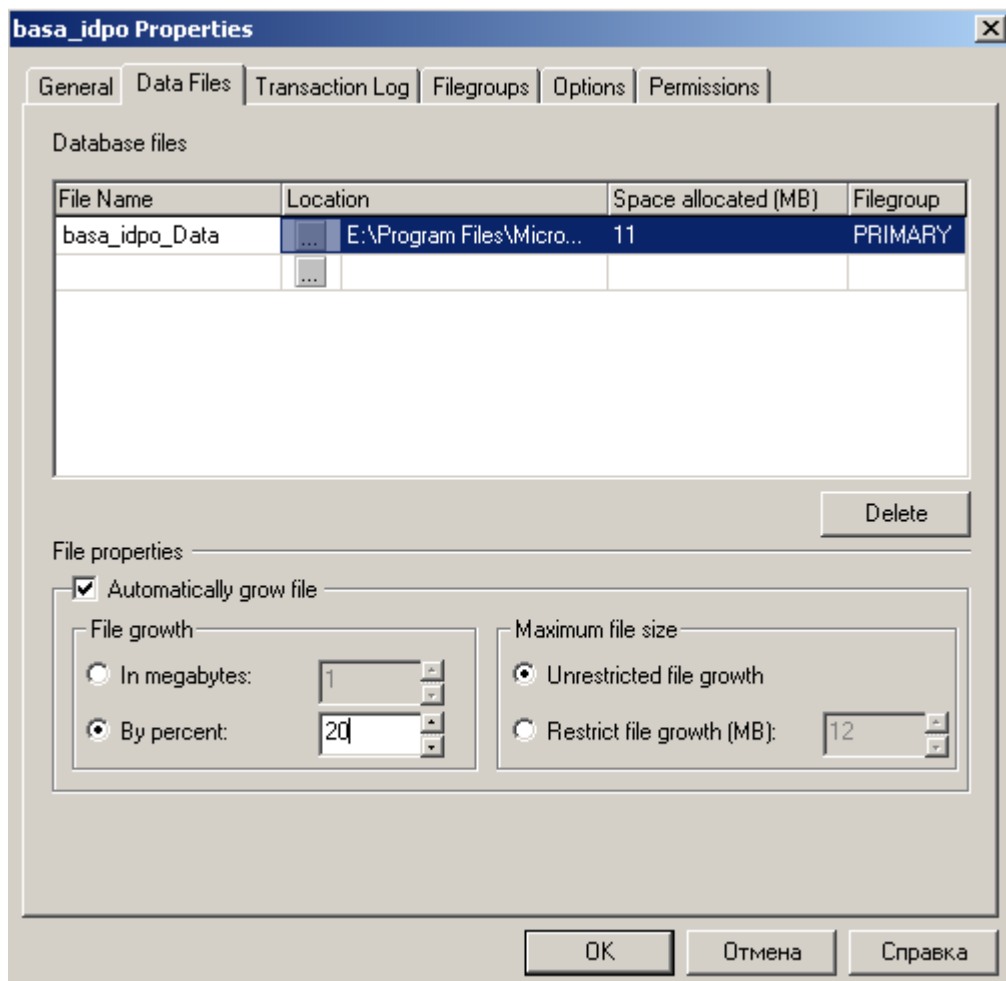
## Установка свойств базы данных.

Когда вы создаете базу данных с помощью мастера Create Database Wizard, вы указываете определенные характеристики, или свойства, базы данных, такие как имя базы данных и место размещения. После создания базы данных вы можете поменять эти свойства, изменив соответствующие параметры в диалоговом окне Properties (Свойства).

Например, увеличение размера физического файла — это довольно серьезная операция, выполнение которой может привести к увеличению времени отклика сервера. Если вы обнаружите, что SQL Server приходится слишком часто увеличивать размер файла. Вам следует учитывать возможность изменения процента увеличения размера файла в диалоговом окне Properties (Свойства), чтобы сервер смог увеличивать размер файла более чем на 10% (установка по умолчанию).

### Изменение процента увеличения размера файла.

1. Выберите свою базу данных в дереве консоли Console Tree.
2. Нажмите кнопку Properties (Свойства) в панели инструментов. SQL Server отобразит диалоговое окно Properties (Свойства) для базы данных.
3. Откройте вкладку Data Files (Файлы данных). SQL Server отобразит свойства файлов данных базы данных.
4. Установите процент увеличения 20%.



5. Нажмите OK. SQL Server установит новое свойство и закроет диалоговое окно Properties (Свойства).

## 1.2. Создание таблиц в SQL Server.

Разработайте следующие таблицы, используя Enterprise Manager (рис.1):

### студенты

	Column Name	Data Type	Length	Allow Nulls
🔑	id_студент	int	4	
	номер_дела	varchar	20	✓
	фамилия	varchar	20	✓
	имя	varchar	20	✓
	отчество	varchar	20	✓
	FK_специальность	int	4	✓
	FK_курс	int	4	✓
	FK_поток	int	4	✓
	место_работы	varchar	100	✓
	год_рождения	smalldatetime	4	✓
	соц_положение	varchar	15	✓
	адрес	varchar	80	✓
	образование	varchar	100	✓
	приказ_зачисления	varchar	30	✓
	дата_значения	smalldatetime	4	✓
	основание_зачисления	varchar	60	✓
	приказ_отчисления	varchar	30	✓
	дата_отчисления	smalldatetime	4	✓
	причина_отчисления	varchar	60	✓
	приказ_восстановления	varchar	30	✓

### дисциплины

	Column Name	Data Type	Length	Allow Nulls
🔑	id_дисциплины	int	4	
	FK_преподаватель	int	4	✓
	FK_предмет	int	4	✓
	FK_специальность	int	4	✓

### учебный план

	Column Name	Data Type	Length	Allow Nulls
🔑	id_уч_план	int	4	
	шифр_спец	varchar	50	✓
	FK_специальность	int	4	✓
	FK_дисциплина	int	4	✓
	FK_курсовая	int	4	✓
	FK_контрольная	int	4	✓
	FK_форма_контр	int	4	✓
	шифр_предмет	varchar	20	✓
	итого	int	4	✓
	лекции	int	4	✓
	практика	int	4	✓
	сам_работа	int	4	✓
	семестр	varchar	10	✓
	номер	int	4	✓



### курс

	Column Name	Data Type	Length	Allow Nulls
🔑	id_курс	int	4	
	курс	varchar	10	✓

### оценка

	Column Name	Data Type	Length	Allow Nulls
🔑	id_оценка	int	4	
	оценка	varchar	20	✓

### контрольная

	Column Name	Data Type	Length	Allow Nulls
🔑	id_кп	int	4	
	контрольная	varchar	15	✓

### поток

	Column Name	Data Type	Length	Allow Nulls
🔑	id_поток	int	4	
	поток	varchar	10	✓

### ф\_контроля

	Column Name	Data Type	Length	Allow Nulls
🔑	id_ф_контроля	int	4	
	контроль	varchar	50	✓

Рис.1.1. Таблицы базы данных

### 1.3. Создание связей между таблицами (рис.1.2)

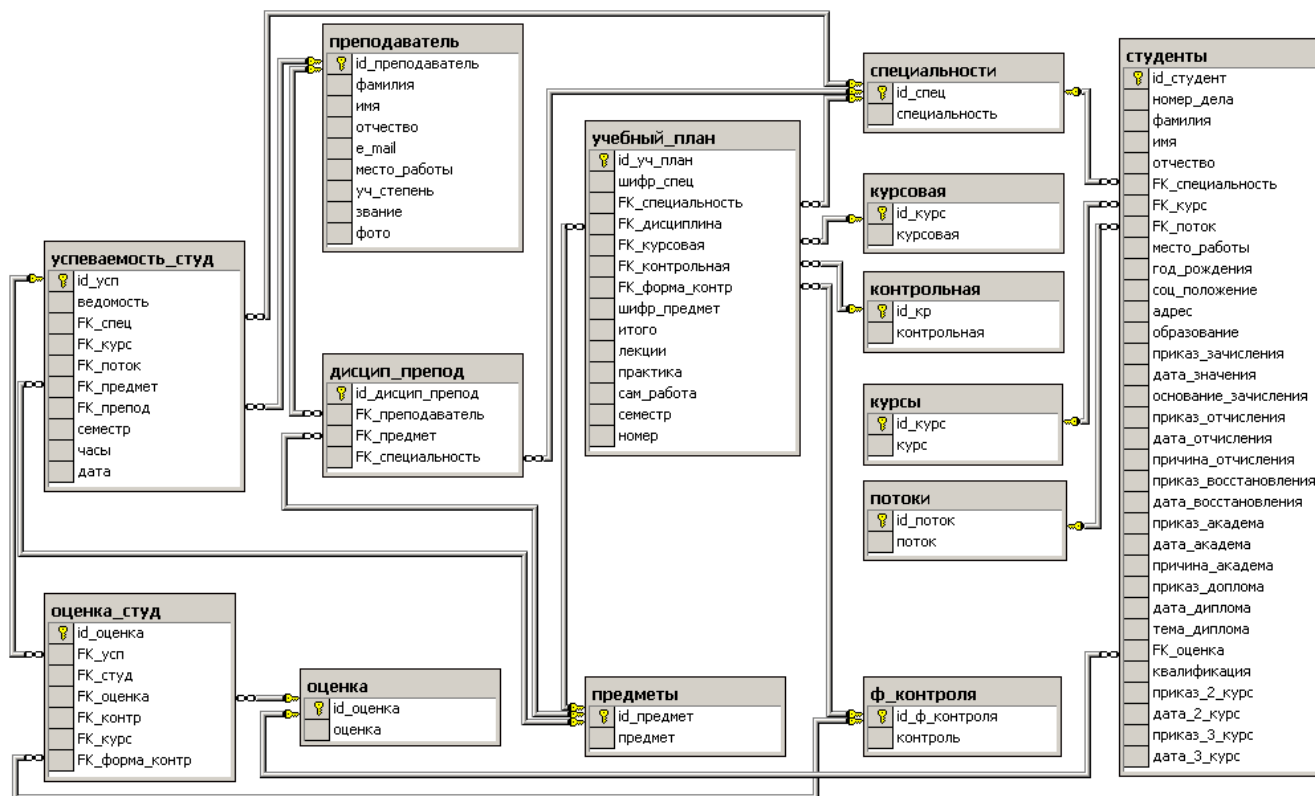


Рис.1.2. Схема базы данных

Таблицы связаны по следующим полям:

- 1- таблица «преподаватель», поле «id\_преподаватель» – таблица «успеваемость\_студ», поле «FK\_препод»;
- 2- таблица «преподаватель», поле «id\_преподаватель» – таблица «дисциплин\_препод», поле «FK\_преподаватель»;
- 3- таблица «успеваемость\_студ», поле «id\_усп» – таблица «оценка\_студ», поле «FK\_усп»;

- 4- таблица «предметы», поле «id\_ предмет» – таблица «учебный\_план», поле «FK\_дисциплина»;
- 5- таблица «предметы», поле «id\_ предмет» – таблица «дисцип\_препод», поле «FK\_предмет»;
- 6- таблица «специальности», поле «id\_ спец» – таблица «учебный\_план», поле «FK\_специальность»;
- 7- таблица «специальности», поле «id\_ спец» – таблица «дисцип\_препод», поле «FK\_специальность»;
- 8- таблица «специальности», поле «id\_ спец» – таблица «студенты», поле «FK\_специальность»;
- 9- таблица «потоки», поле «id\_ поток» – таблица «студенты», поле «FK\_поток»;
- 10- таблица «курсы», поле «id\_ курс» – таблица «студенты», поле «FK\_курс»;
- 11- таблица «оценка», поле «id\_ оценка» – таблица «студенты», поле «FK\_оценка»;
- 12 - таблица «предметы», поле «id\_ предмет» – таблица «успеваемость\_студ», поле «FK\_предмет»;
- 13 - таблица «специальности», поле «id\_ спец» – таблица «успеваемость\_студ», поле «FK\_спец»;
- 14- таблица «оценка», поле «id\_ оценка» – таблица «оценка\_студ», поле «FK\_оценка».

#### 1.4. Создание триггеров на SQL Server.

Создайте триггеры, включающие в себя транзакции по откату неверно выполненных операций для таблиц успеваемость\_студ и оценка\_студ.

## 1.5. Ввод первоначальных данных на сервере.

Заполните следующие словари:

- таблица «преподаватель» - введите 4 записи по полям «фамилия», «имя», «отчество», «телефон»;
- таблица «предметы» – 5 предметов;
- таблица «специальности» - 5 записей (5 специальностей);
- таблица «потоки» - 3 записи:

	id поток	поток
▶	1	1 поток
	2	2 поток
	3	3 поток
*		

- таблица «курсы» - 3 записи:

	id курс	курс
▶	1	1 курс
	2	2 курс
	3	3 курс
*		

- таблица «оценка» - 4 записи:

	id оценка	оценка
▶	1	отлично
	2	хорошо
	3	удовлетворительно
	4	неудовлетворительно

- таблица «ф\_контроля» - 4 записи, 1 запись пустая:

	id ф_контроля	контроль
▶	1	
	2	зачет
	3	экзамен
	4	зачет-экзамен

- таблица «контрольная» - 2 записи, 1 запись пустая:

	id_кр	контрол_р
▶	1	
	2	контрольная работа

- таблица «курсовая» - 6 записей, 1 запись пустая:

	id_курс	курс_р
	1	
	2	курсовая работа
	3	отлично
	4	хорошо
	5	удовлетворительно
	6	неудовлетворительно