МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Московский Авиационный институт

(Национальный исследовательский университет)

Институт №8

«Информационные технологии и прикладная математика»

Кафедра 813

«Компьютерная математика»

Курсовой проект по дисциплине «Базы данных»

Тема:

«Проектирование реляционных баз данных»

ВАРИАНТ 3

Студент: Евкарпиев Михаил Димитриевич

Группа: М8О-З 11Б-19

Преподаватели: Денисова И.П. Чумакова Е.В.

Мокряков А.В.

Оценка: Дата:

Москва

2021

Оглавление

[ВВЕДЕНИЕ 3](#_Toc91082656)

[1. Анализ предметной области и постановка задачи 3](#_Toc91082657)

[1.1. Описание предметной области и назначение системы 3](#_Toc91082658)

[1.2. Цели разработки и способы Достижения Цели разработки: 4](#_Toc91082659)

[1.3. Ожидаемые результаты 4](#_Toc91082660)

[1.4. Информационный образ системы 4](#_Toc91082661)

[1.5. Входные и ВЫХОДНЫе потоки 5](#_Toc91082662)

[1.6. Функциональный образ системы 5](#_Toc91082663)

[1.7. Требования и ограничения на использование системы 6](#_Toc91082664)

[2. Концептуальное и логическое проектирование базы данных 6](#_Toc91082665)

[2.1. Выделение информационных объектов и определение атрибутов объектов 6](#_Toc91082666)

[2.2. Разработка логической структуры базы Данных 8](#_Toc91082667)

[2.3. Построение Диаграммы сущность-связь 9](#_Toc91082668)

[2.3.1. Раскрытие связей многие ко многим 9](#_Toc91082669)

[2.3.2. Поддержка ограничений целостности, списки Доменов атрибутов 10](#_Toc91082670)

[2.3.3. Нормализация объектной модели 10](#_Toc91082671)

[2.4. Выбор используемой СУБД 11](#_Toc91082672)

[3. Физическое проектирование базы данных 11](#_Toc91082673)

[3.1. Генерация физической модели БД 11](#_Toc91082674)

[3.2. Генерация физической схемы БД 12](#_Toc91082675)

[3.3. Описание запросов и процедур к БД 13](#_Toc91082676)

[ЗАКЛЮЧЕНИЕ 15](#_Toc91082677)

[СПИСОК ЛИТЕРАТУРЫ 16](#_Toc91082678)

# ВВЕДЕНИЕ

В последние годы бурное развитие новых технологий сделали актуальным использование баз данных в организациях различных форм собственности и направлений деятельности. Успехи в исследованиях и разработке баз данных стали основой фундаментальных разработок коммуникационных систем, транспорта и логистики, финансового менеджмента, методов доступа к научной литературе, а также многих гражданских и военных приложений.

Базы данных лежат в основе информационных систем, и это коренным образом изменило характер работы многих предприятий и организаций. Цель базы данных – помочь людям и организациям вести учет, осуществлять информационную поддержку при принятии решений. Базы данных представляют собой особую организацию данных и предполагают использование специальных программно-языковых средств. Базы данных - это сложные взаимосвязанные информационные образования, предназначенные для многоцелевого использования.

# Анализ предметной области и постановка задачи

## Описание предметной области и назначение системы

Разрабатываемая подсистема носит название ”Деканат” и предназначена для:

* + автоматизации обработки информации о Студентах и Преподавателях;
  + обеспечения возможности оперативного поиска студента/преподавателя по заданным параметрам;
  + обеспечения возможности оперативного получения информации о сессии.

База данных создаётся для информационного обслуживания сотрудников деканата и студентов. Данная система понадобится сотрудникам деканата для автоматизации и облегчения процедуры поиска информации о студентах/преподавателях. БД должна содержать данные об студентах, преподавателях, сессии, спезиализациях и предметах.

В соответствии с предметной областью система строится с учётом следующих особенностей:

* каждый студент учится в определённой группе, на определенной кафедре и факультете, в каждой группе учатся несколько студентов - каждый студент сдает сессию, по каждому студенту может содержаться дополнительная информация.

## Цели разработки и способы Достижения Цели разработки:

* обеспечить минимальные затраты времени для получения информации;
* упростить поиск нужном студенте/преподавателе;
* обеспечить оперативность изменения информации.

Способы Достижения:

Указанные цели разработки могут быть достигнуты следующими способами:

* + проанализировать документы, которые имеют отношение к решаемой задаче для выявления значимых объектов (сущностей) и атрибутов сущностей — данных, которые должны храниться в БД
  + описать модель данных в удобном для разработчиков способе (в виде ЕК—диаграммы)
  + представить информацию в связанных таблицах
  + разработать запросы и процедуры для автоматического учета студентов и преподавателях;

## Ожидаемые результаты

В результате применения разрабатываемой системы ожидается:

* упрощение работы сотрудникам деканата, работающих в администрации;
* формирование необходимой информации не более чем за 1мин., вместо n мин. при ручном поиске;

## Информационный образ системы

Для создания ER-моделей необходимо выделить сущности предметной области:

|  |  |  |  |
| --- | --- | --- | --- |
| п/п | Наименование | Определение | |
| 1 | Faculty | Факультет | |
| 2 | Specialization |  | Название специализаций |
| 3 | Group\_st | Группа студентов | |
| 4 | Additional\_information | Дополнительная информаци о студентах | |
| 5 | Discipline | Содержит наименование дисциплин | |
| 6 | Employees | Сотрудники кафедры/преподаватели | |
| 7 | Kafedra | Кафедра | |
| 8 | List\_of\_expelled | Содержит список отчисленных студентов | |
| 9 | Scientific\_degrees | Содержит список возможных научных степеней | |
| 10 | Student | Основная информация о студенте | |
| 11 | Subject | Предмет с указанным названием | |
| 12 | Vedomost | Ведомость, содержит информацию о сессии | |

## Входные и ВЫХОДНЫе потоки

Входные:

Список обучающихся

|  |  |  |  |
| --- | --- | --- | --- |
| Индификатор студента | Дата рождения | Номер зачетной книжки | Адрес |
|  |  |  |  |

Данные о студенте

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ФИО | Факультет | Кафедра | Группа | Специализация | Курс | Статус |
|  |  |  |  |  |  |  |

Выходные:

- Данные студентов;   
- Информация о направлении обучения;   
- Информация о статусе обучающихся;

## Функциональный образ системы

Система предназначена для решения следующих задач:

1. Выдавать информацию о студентах находящихся в академическом отпуске;
2. Выдавать информацию о студентах обучающихся по направлению Прикладная информатика

З. Выдавать информацию о студентах сдававших экзамен преподавателю Голубев;

1. Выдавать количество отчисленных студентов;
2. Выдавать информацию о студентах принадлежащих кафедре Вычислительные машины, системы и сети;
3. Выдавать сотрудников со степенью не ниже доктора наук;
4. Выдавать группы, их специализации и FIO студента в которых есть студенты в академическом отпуске;

## Требования и ограничения на использование системы

Пользователи системы:

Данная система является вспомогательной (справочной) системой и поэтому может использоваться администратором и всеми сотрудниками деканата и студентами, а также входить в состав различных других систем

Администратор имеет полный доступ к базе. Остальные сотрудники деканата могут:

* Добавлять обновлять информацию о себе или студентах
* Получать информацию о студентах и других сотрудниках
* Заполнять ведомости

Студенты:

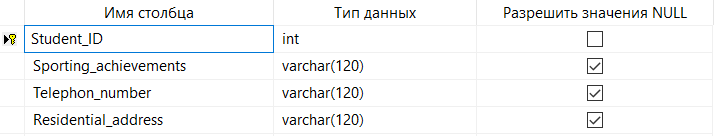
* Обновлять дополнительную информацию о себе
* Просматривать информацию о студентах и преподавателях

# Концептуальное и логическое проектирование базы данных

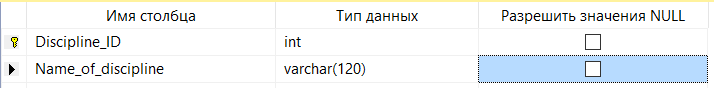
## Выделение информационных объектов и определение атрибутов объектов

Для каждого сущности из п. 1.4. определим первичный ключ (служащий для идентификации записей) отношения и перечень атрибутов, имеющих значение в предметной области задачи. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной:

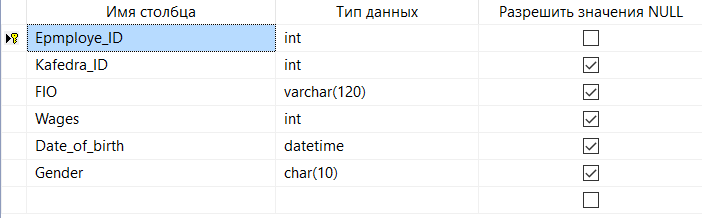
Additional\_information



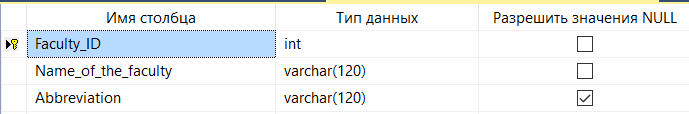
Discipline



Employees



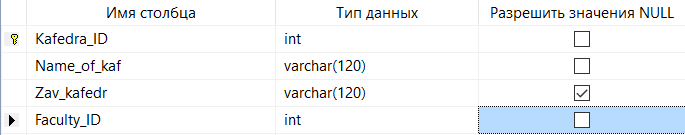
Faculty



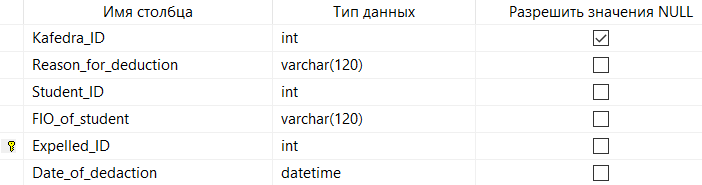
Group\_st



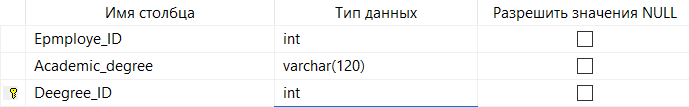
Kafedra



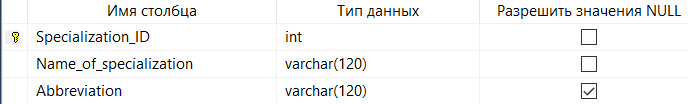
List\_of\_expelled



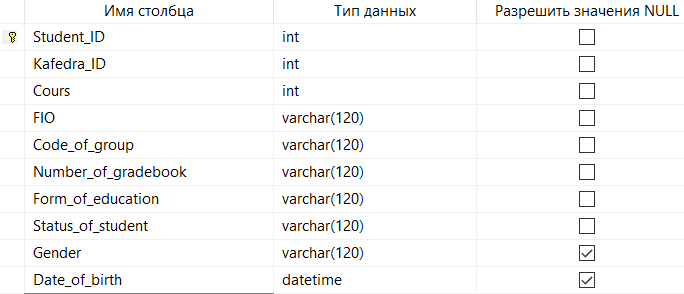
Scientific\_degrees



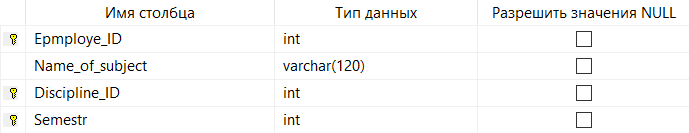
Specialization



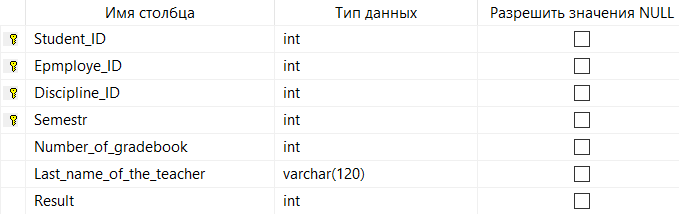
Student



Subject



Vedomost



## Разработка логической структуры базы Данных

Для построения ЕК-модели на основе выделенных сущностей, необходимо определить типы связей между ними. Erwin Data Modeler поддерживает три типа связей (или отношений): один-ко многим, один-к-одному, многиеко-многим.

Связь типа один-ко-многим — межтабличное отношение, при котором любая запись в первой таблице может быть связана несколькими записями во второй, но в то же время любая запись второй таблицы связана только с одной записью в первой.

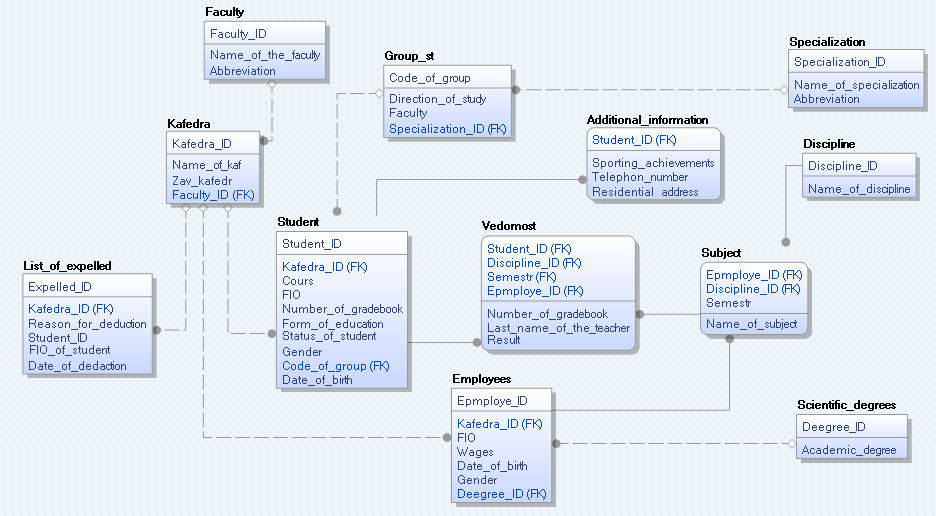
Например, связь между таблицами « Faculty » и « Kafedra » в базе данных «Деканат» — связь типа один-ко-многим, т.к. один факультет может иметь несколько кафедр, но кафедра принадлежит одному факультету.

Если при этом связующее поле в одной из таблиц является ключевым, то такая таблица называется главной (таблица « Faculty »).

Вторая таблица, участвующая в связи, называется подчиненной (таблица « Kafedra »). При этом связующее поле подчиненной таблицы обычно называют внешним (или чужим) ключом.

Связь многие-ко-многим установлена между таблицами Student-Subject, а также между таблицами Employees-Discipline. Все остальные связи в базе данных «Деканат»- это связи типа один-ко-многим.

В результате была получена ER-моделей приведенная на рисунке 1.

  
Рис. 1. Логическая структура Базы данных

## Построение Диаграммы сущность-связь

### 2.3.1. Раскрытие связей многие ко многим

В полученной ЕК-модели имеется 2 связи многие-ко-многим, которые не поддерживаются ни одной из реляционной СУБД. Подобные связи преобразуются (раскрываются) через промежуточную таблицу. Таким образом, в модель будут добавлены три сущности, каждая из которых является подчинённой по отношению к таблицам, объединённым связью многие-комногим (т.е. связывается с этими таблицами связью многие-к-одному). При этом каждое из отношений будет иметь составной первичный ключ и двух внешних ключей главных таблиц.

Employees и Discipline раскрывается таблицей Subject.

Отношение, описывающее предметы, которые ведет преподаватель. В данном случае такая связь необходима из-за того, что преподаватель может вести предметы сразу по нескольким дисциплинам, также как и дисциплину могут вести несколько преподавателей.

Student и Subject раскрывается таблицей Vedomost.

Здесь связь многие ко многим обусловлена тем, что у одного студента может быть несколько сдаваемых предметов, а у одного предмета может быть несколько студентов сдающих его.

### 2.3.2. Поддержка ограничений целостности, списки Доменов атрибутов

Для сохранения целостности данных определим следующие ограничения целостности:

* обязательные данные;
* ограничения для доменов атрибутов;
* целостность сущностей;
* ссылочная целостность.

При оправлении атрибутов сущностей были указаны обязательные для заполнения данные, т.е. этим атрибуты не могут иметь пустого (NULL) значения, для этого в свойствах атрибута в Erwin Data Modeler зададим обязательность (NOT NULL).

Данные ограничения устанавливаются при определении доменов атрибутов, присутствующих в модели данных

Первичный ключ любой сущности не может содержать пустого значения, для этого в свойствах атрибута в Erwin Data Modeler зададим обязательность (NOT NULL).

Внешний ключ связывает каждую строку дочернего отношения с той строкой родительского отношения, которая содержит это же значение соответствующего первичного ключа. Понятие ссылочной целостности означает, что если внешний ключ содержит некоторое значение, то оно обязательно должно присутствовать в первичном ключе одной из строк родительского отношения.

### 2.3.3. Нормализация объектной модели

Хранение информации в реляционной БД, предполагает такую структуру, в которой данные хранятся в оптимальном объеме. Для этого предусмотрен механизм нормализации. Проверим на соответствие как минимум З нормальной форме полученных отношений диаграммы.

Все таблицы в данной системе обладают следующими свойствами:

1. Каждый элемент таблицы представляет 1 элемент данных (атомарность).
2. Все столбцы таблиц однородные
3. Отсутствуют повторяющиеся атрибуты (соответствие 1 НФ)
4. Отсутствуют атрибуты, зависящие только от части уникального идентификатора (2НФ)
5. Отсутствуют атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор (ЗНФ), т.е. все отношения находится в З нормальной форме, т.к. находятся во 2НФ и каждый не ключевой атрибут не транзитивно зависит от первичного ключа

## Выбор используемой СУБД

Анализ информационных задач показывает, что для реализации требуемых функций подходят почти все (MS Access, MySQL, MS SQL Server и др.). Все они поддерживают реляционную модель данных и предоставляют разнообразные возможности для работы с данными.

В Erwin Data Modeler доступно генерировать физическую базу данных на основе разработанной логической модели в широкий спектр реляционных СУБД (с том числе во все перечисленные).

Для выполнения курсового работы была выбрана СУБД MS SQL Server. MS SQL Server — сервер баз данных, реализует подход ”клиент-сервер”, главными достоинствами которого являются высокая степень защиты данных, мощные средства для обработки данных, высокая производительность. Область применения: хранение больших объемов данных, хранение высокоценных данных или данных, требующих соблюдения режима секретности.

# Физическое проектирование базы данных

## Генерация физической модели БД

В результате проведенной разработки получили ER-модель физической модели БД, которая приведена на рисунке 2.

В Erwin Data Modeler выбираем тип СУБД, в которой необходимо создать физическую базу данных. При этом:

* + каждая простая сущность превращается в таблицу, имя сущности становится именем таблицы.
  + каждый атрибут становится возможным столбцом с тем же именем; может выбираться более точный формат, столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения;
  + компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы.

В итоге формируется sql-скрипт содержащий запросы на создание базы данных и таблиц с определенными связями и ограничениями целостности.

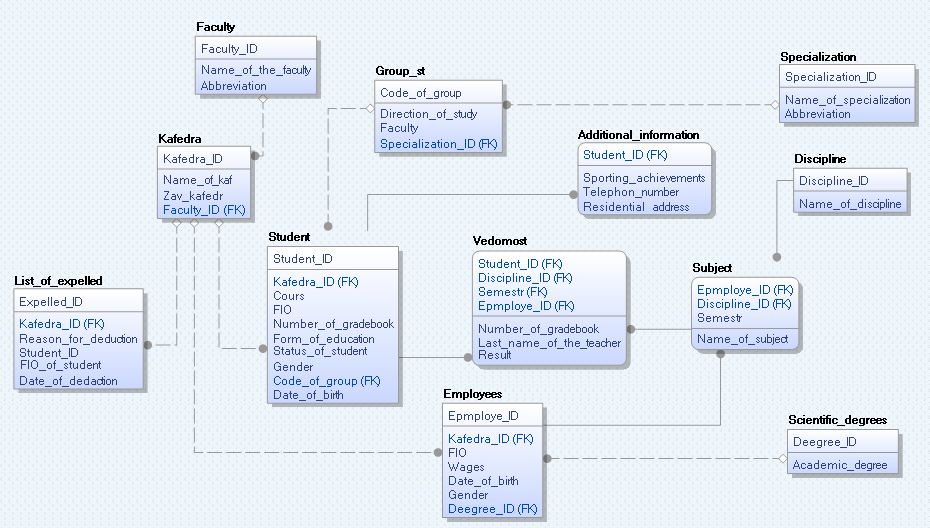


Рис.2. Физическая модель базы данных

## Генерация физической схемы БД

Результатом выполнения sql-скрипт является физическая база данных, схема данных которой представлена на рисунке З.

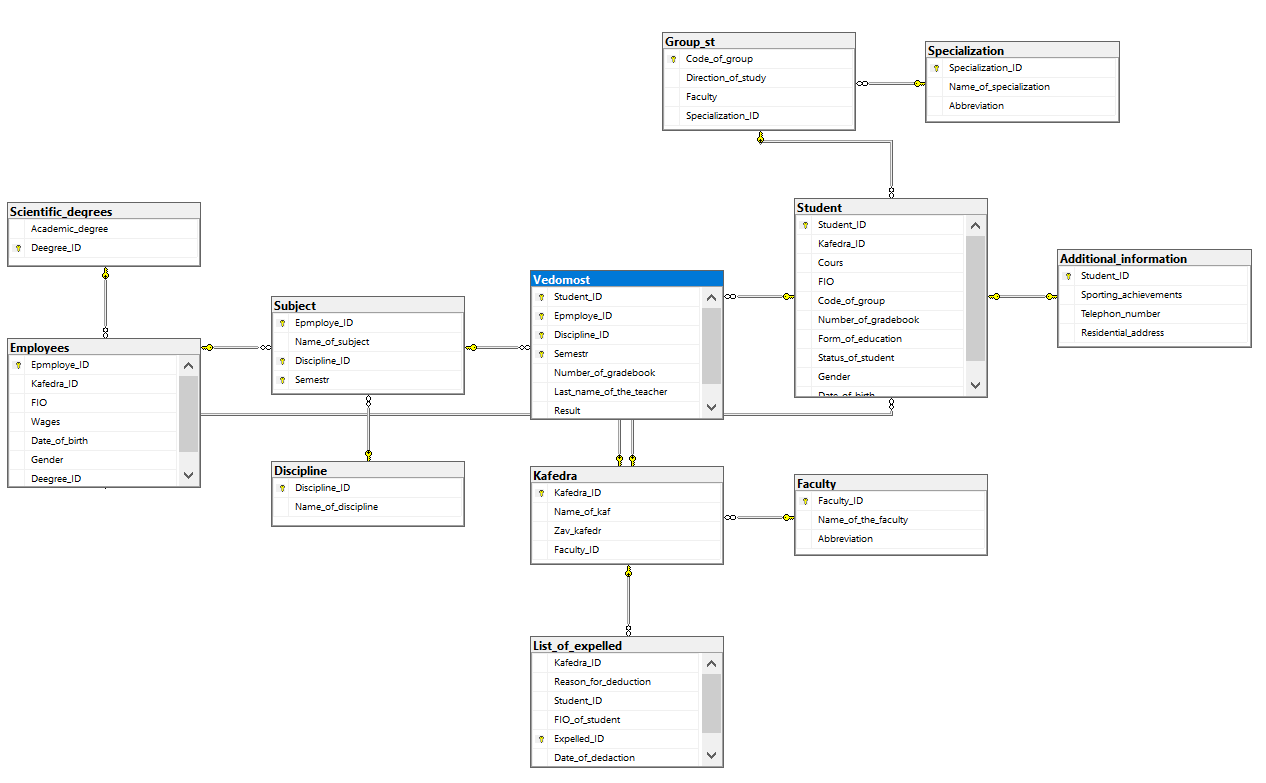


Рис.3. Физическая схема базы данных

## Описание запросов и процедур к БД

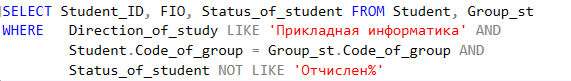
Для демонстрации придем примеры реализации основных запросов, реализующих основные заявленные функции:

* Вывести всех студентов находящихся в акадаемическом отпуске





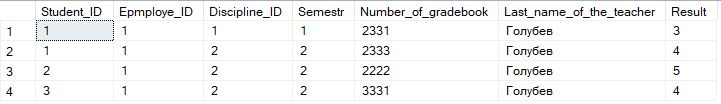
* Вывести список студентов обучающихся по направлению Прикладная информатика





* Вывести студентов сдававших экзамен преподавателю Голубев



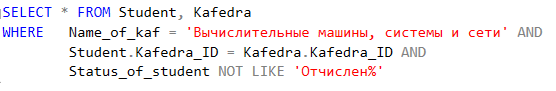


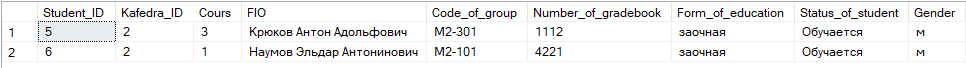
* Определить кол-во отчисленных студентов



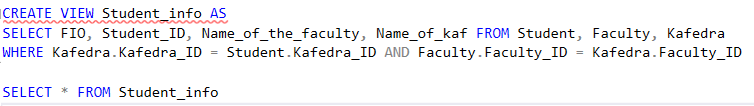


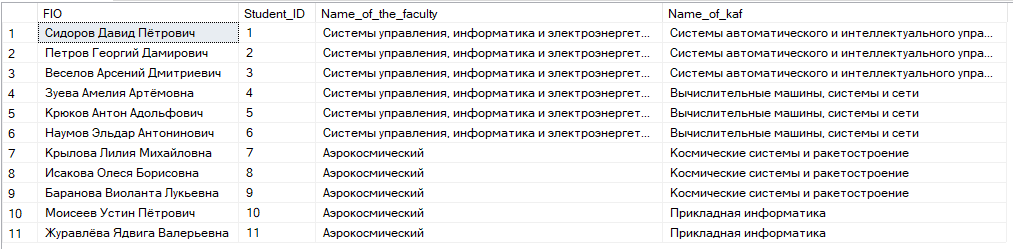
* Вывести список студентов принадлежащих кафедре Вычислительные машины, системы и сети



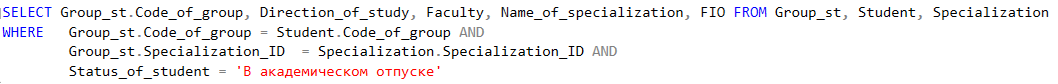


* Вывести представление в котором содержится информация о студенте(ФИО и ID), факультете и кафедре на которых он учится.





* Вывести группы, их специализации и FIO студента в которых есть студенты в академическом отпуске



Опишем для всех групп пользователей права доступа к каждой таблице

* S – чтение данных (Select)
* I – добавление данных (insert)
* U – модификация данных (update)
* D – удаление данных (delete)

|  |  |  |  |
| --- | --- | --- | --- |
| Таблицы\Группы пользователей | Администратор | Сотрудник деканата | Студент |
| Additional\_information | SIUD | S | SIUD |
| Discipline | SIUD | S | S |
| Employees | SIUD | SU | S |
| Faculty | SIUD | S | S |
| Group\_st | SIUD | S | S |
| Kafedra | SIUD | S | S |
| List\_of\_expelled | SIUD | S | S |
| Scientific\_degrees | SIUD | SU | S |
| Specialization | SIUD | S | S |
| Student | SIUD | S | S |
| Subject | SIUD | S | S |
| Vedomost | SIUD | SIUD | S |

Права назначает администратор БД

Права доступа предоставляются с помощью команды GRANT. В качестве примера возьмем права доступа сотрудника деканата (deacanat\_empl). Права доступа к отношениям Vedomost и Student могут быть описаны следующим образом:  
 grant select, insert, update, delete on Vedomost to decanat\_empl;  
 grant select on Student to decanat\_empl;

# ЗАКЛЮЧЕНИЕ

Таким образом, работе было показано, как создать базу данных «Деканата» для повышения эффективности работы администрации деканата. Также были реализованы возможности:

* поиска студентов и преподавателей по различным параметрам;
* просмотра дополнительной информации о студентах;

И функциональные возможности:

1. Выдавать информацию о студентах по статусу обучения, направлению обучения, сдаче определенному экзамена/зачета/рейтинга определенному преподавателю, принадлежности к кафедре.
2. Определять количество отчисленных студентов, сотрудников, имеющих определенную научную степень.
3. Выводить группы, в которых есть студенты, находящиеся в академическом отпуске и ФИО этих студентов.

# СПИСОК ЛИТЕРАТУРЫ

1. Баженова И.Ю. Основы проектирования приложений баз данных (2е изд.) - М.: НОУ ”Интуит”, 2016.- 237с.

1. https://metanit.com/sql/sqlserver/ 2.

З. https://www.erwin.com/solutions/data-modeling/