Белорусский государственный технологический университет

факультет информационных технологий

кафедра информационных систем и технологий

**Методы сбора, хранения, обработки и анализа данных**

**Лабораторная работа №1**

Разработал Лопатнюк П.В.

ФИТ, 3 курс, группа 1/1

Проверил Нистюк О. А.

Минск 2024

# Проектирование базы данных

## 1.1. Список типичных бизнес-задач

Типичными бизнес-задачами являются управление данными о студентах, преподавателях, предметах, группах, а также оценка их успеваемости и распределение стипендий. Ведение данных о студентах включает хранение информации об их имени, дате рождения, годе поступления, принадлежности к группе, наличии стипендии и адресе проживания. Помимо этого, важно управлять назначением стипендий в зависимости от статуса студента (бюджет/контракт) и их успеваемости, а также распределять студентов по предметам в зависимости от группы и семестра.

Задачи по мониторингу успеваемости студентов включают хранение истории оценок по каждому предмету, расчёт средней успеваемости для определения стипендий, а также анализ успеваемости по предметам и семестрам. Преподавательский состав управляется путём ведения данных о каждом преподавателе, включая его роль и департамент, а также назначением преподавателей на конкретные предметы и привязкой их к факультетам и департаментам.

Управление предметами и группами включает создание и поддержку списка учебных предметов, их типа (экзамен или зачёт), а также назначение этих предметов на определённые группы в зависимости от семестра. Факультеты и департаменты также играют важную роль: ведётся информация о них, преподаватели назначаются на соответствующие факультеты и департаменты, а также поддерживаются связи между этими подразделениями.

Также важной задачей является анализ данных и формирование отчётов, включая отчёты по успеваемости студентов, распределению стипендий и данные для административной отчётности. Наконец, обеспечение безопасности данных требует управления доступом для различных пользователей системы (администраторов, преподавателей, студентов), а также соблюдения конфиденциальности данных, включая информацию о студентах и их успеваемости.

## 1.2. Список пользователей проекта

Список пользователей проекта включает следующие категории:

* Администрация университета. Это могут быть сотрудники деканатов, учебных отделов или других административных подразделений, которые используют базу данных для анализа успеваемости студентов, управления учебным процессом, назначения стипендий, формирования отчётов и подготовки документации.
* Преподаватели. Преподаватели используют систему для ведения и анализа информации по своим предметам. Они могут просматривать успеваемость студентов, вносить оценки, а также получать доступ к информации о группах, к которым они прикреплены.
* Студенты. Студенты могут использовать систему для просмотра информации о своих оценках, успеваемости, назначенных предметах и группах. Также они могут получать информацию о стипендиях и других аспектах, связанных с их академической деятельностью.
* Отделы управления персоналом. Сотрудники отдела кадров используют базу данных для управления данными о преподавателях и административном персонале, их распределении по факультетам и департаментам, а также для формирования отчётности.
* Финансовый отдел. Сотрудники финансового отдела используют систему для управления и мониторинга распределения стипендий, финансовых выплат, а также анализа финансовых аспектов, связанных с бюджетом студентов и преподавателей.

# 2. Проверка логической схемы

## 2.1. Применение нормализации до 4 НФ

Нормализация – это процесс структурирования данных в базе данных для уменьшения избыточности и предотвращения аномалий при внесении изменений в данные. Процесс нормализации проходит через несколько этапов, называемых нормальными формами (НФ), каждая из которых накладывает определённые требования на структуру таблиц. В проекте базы данных университета нормализация будет применена следующим образом:

* Первая нормальная форма (1 НФ): Все атрибуты должны быть атомарными, т.е. не должны содержать составных или повторяющихся групп.
* Вторая нормальная форма (2 НФ): Требует, чтобы все неключевые атрибуты зависели от полного первичного ключа, а не от его части. Это исключит избыточные данные, такие как хранение факультета в таблице студентов, если он уже хранится в таблице группы.
* Третья нормальная форма (3 НФ): Все неключевые атрибуты должны зависеть только от первичного ключа и не должны зависеть от других неключевых атрибутов.
* Четвёртая нормальная форма (4 НФ): Устраняет многозначные зависимости, то есть ситуации, когда одно поле связано с двумя и более наборами значений другого поля.

Таким образом, нормализация до 4 НФ гарантирует отсутствие избыточности и излишних зависимостей в данных, что приводит к повышению производительности и целостности данных в системе.

## 2.2. Анализ необходимости денормализованных атрибутов

Денормализация — это обратный процесс нормализации, при котором данные умышленно дублируются в некоторых таблицах для повышения производительности системы за счёт снижения сложности запросов.

В данном проекте в денормализации необходимости нет.

## 2.3. Обеспечение безопасности данных

Обеспечение безопасности данных в проекте базы данных университета является ключевым аспектом, учитывая конфиденциальность данных студентов, преподавателей и академических результатов. Безопасность данных будет обеспечиваться на нескольких уровнях:

* Контроль доступа: для разных категорий пользователей (администраторы, преподаватели, студенты, финансовый отдел) будут определены различные уровни доступа к данным.
* Аутентификация и авторизация: пользователи будут идентифицироваться с помощью уникальных логинов и паролей.
* Шифрование данных: Для защиты конфиденциальных данных, таких как оценки студентов, личные данные и финансовая информация, будут использоваться механизмы шифрования данных как на уровне хранения, так и при передаче данных по сети.
* Резервное копирование: Чтобы минимизировать риск потери данных из-за сбоев, будут реализованы регулярные процессы резервного копирования базы данных.

# 3. Оценка сроков и ресурсов проекта

## 3.1. Параметры оценки времени выполнения

Оценка времени выполнения проекта по созданию базы данных университета основывается на нескольких ключевых параметрах:

Анализ и проектирование: Время выполнения зависит от сложности структуры данных и может занять от 3 до 4 недель.

Нормализация данных: На это может потребоваться около 1 недели, в зависимости от количества таблиц и их взаимосвязей.

Создание таблиц в SQL Server и Oracle: После проектирования структуры базы данных необходимо реализовать её физическую модель. Время создания таблиц и написания соответствующих SQL-запросов зависит от сложности структуры и может варьироваться от 2 до 3 недель для каждой базы данных. Также потребуется время на подключение к серверам SQL Server и Oracle.

Тестирование и проверка: В зависимости от размера базы данных и объёма данных для тестирования, этот процесс может занять от 2 до 3 недель.

Настройка и развёртывание: После завершения тестирования база данных должна быть развёрнута в рабочей среде, настроена для пользователей, а также должна быть выполнена интеграция с другими системами, если это необходимо. Этот этап может занять до 2 недели.

Таким образом, общее время выполнения проекта может варьироваться от 10 до 17 недель.

## 3.2. Расчёт необходимых ресурсов

Для разработки, тестирования и развертывания базы данных потребуется выделить серверы под SQL Server и Oracle.

Будут использоваться среды разработки и тестирования, такие как SQL Server Management Studio, Oracle SQL Developer, системы контроля версий, а также средства для шифрования данных и защиты соединений (SSL/TLS).

Требуется дисковое пространство для хранения данных, резервных копий и журналов транзакций.

# 4. Структура базы данных

## 4.1. Диаграммы

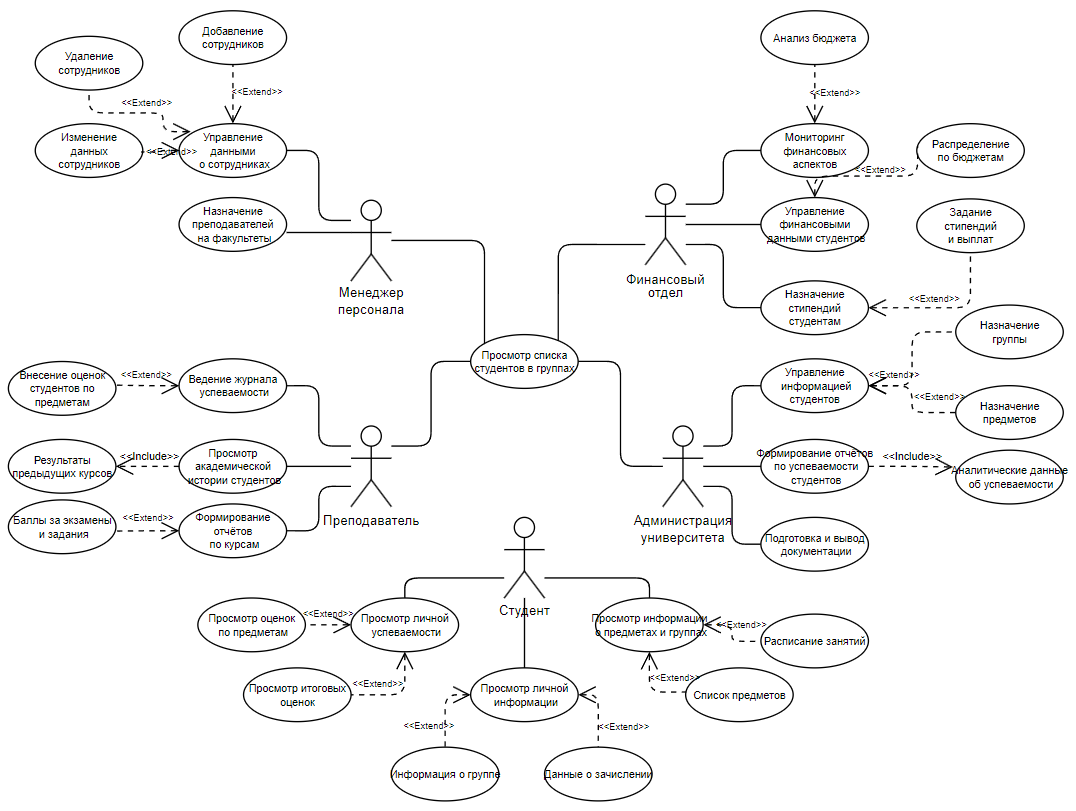


Рисунок 4.1 − UML-диаграмма вариантов использования

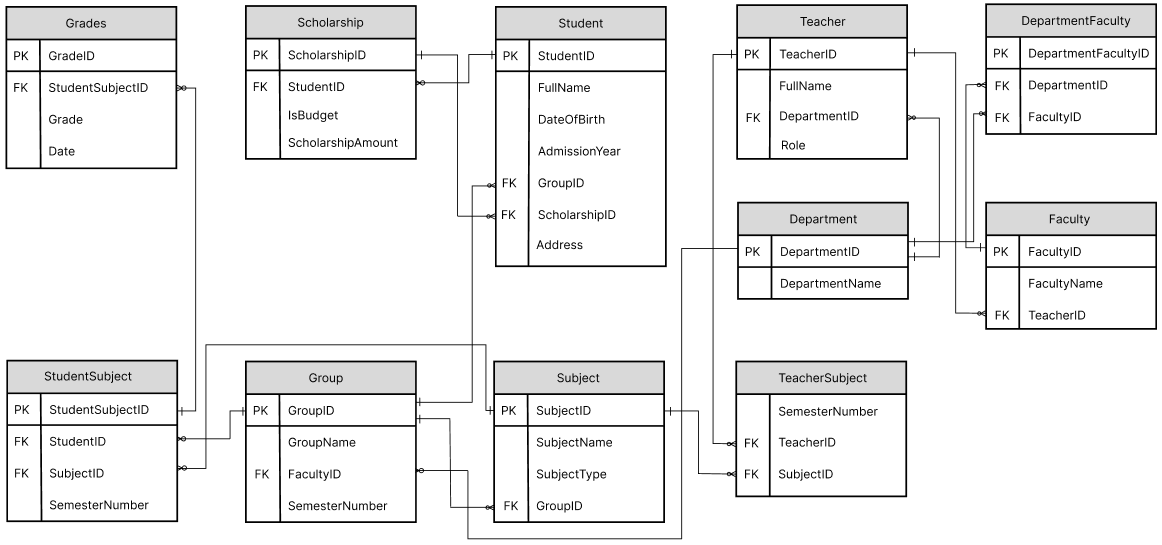


Рисунок 4.2 − Логическа схема БД

Описание связей в логической схеме базы данных университета:

1. Связь между таблицами Student и Group

Тип связи: Один ко многим.

Описание: Один студент может быть зачислен только в одну группу, но в одной группе может находиться много студентов. В таблице Student присутствует внешний ключ GroupID, который ссылается на таблицу Group.

1. Связь между таблицами Group и Faculty

Тип связи: Один ко многим.

Описание: Один факультет может включать в себя несколько групп, но каждая группа относится только к одному факультету. В таблице Group используется внешний ключ FacultyID, ссылающийся на таблицу Faculty.

1. Связь между таблицами StudentSubject и Student

Тип связи: Один ко многим.

Описание: Один студент может изучать много предметов, и по каждому предмету создается запись в таблице StudentSubject. Таблица StudentSubject имеет внешний ключ StudentID, который ссылается на таблицу Student.

1. Связь между таблицами StudentSubject и Subject

Тип связи: Один ко многим.

Описание: Один предмет может изучаться множеством студентов, но каждый студент изучает конкретный предмет. Таблица StudentSubject использует внешний ключ SubjectID, ссылающийся на таблицу Subject.

1. Связь между таблицами Grades и StudentSubject

Тип связи: Один ко многим.

Описание: Для каждого предмета, который студент изучает, может быть выставлено множество оценок. Таблица Grades содержит внешний ключ StudentSubjectID, который связывает оценку с конкретной записью в таблице StudentSubject.

1. Связь между таблицами Scholarship и Student

Тип связи: Один ко многим.

Описание: Один студент может получать только одну стипендию, но у многих студентов могут быть разные стипендии. Таблица Scholarship содержит внешний ключ StudentID, ссылающийся на таблицу Student.

1. Связь между таблицами TeacherSubject и Teacher

Тип связи: Один ко многим.

Описание: Один преподаватель может преподавать несколько предметов. Таблица TeacherSubject использует внешний ключ TeacherID, ссылающийся на таблицу Teacher.

1. Связь между таблицами TeacherSubject и Subject

Тип связи: Один ко многим.

Описание: Один предмет может преподаваться несколькими преподавателями, и для этого создается несколько записей в таблице TeacherSubject. Внешний ключ SubjectID связывает преподавателя с предметом.

1. Связь между таблицами Department и Teacher

Тип связи: Один ко многим.

Описание: Один департамент (кафедра) может включать в себя несколько преподавателей, но каждый преподаватель относится только к одному департаменту. В таблице Teacher используется внешний ключ DepartmentID.

1. Связь между таблицами Faculty и Teacher

Тип связи: Один к одному.

Описание: На каждом факультете может быть один преподаватель, который является деканом или руководителем. Таблица Faculty содержит внешний ключ TeacherID, ссылающийся на таблицу Teacher.

11.Связь между таблицами DepartmentFaculty и Department

Тип связи: Один ко многим.

Описание: Один департамент может быть связан с несколькими факультетами через таблицу DepartmentFaculty, которая имеет внешний ключ DepartmentID, ссылающийся на таблицу Department.

12. Связь между таблицами DepartmentFaculty и Faculty

Тип связи: Один ко многим.

Описание: Один факультет может иметь несколько кафедр, и эта связь осуществляется через таблицу DepartmentFaculty. В таблице используется внешний ключ FacultyID, ссылающийся на таблицу Faculty.

Связи в данной схеме в основном имеют тип один ко многим, что позволяет структурировать данные в иерархическом виде. Это соответствует логике учебного процесса, где каждый студент и преподаватель связан с несколькими учебными и административными сущностями (группы, предметы, департаменты).

Таблицы StudentSubject и TeacherSubject являются связующими таблицами для реализации отношений многие ко многим между студентами/предметами и преподавателями/предметами соответственно.

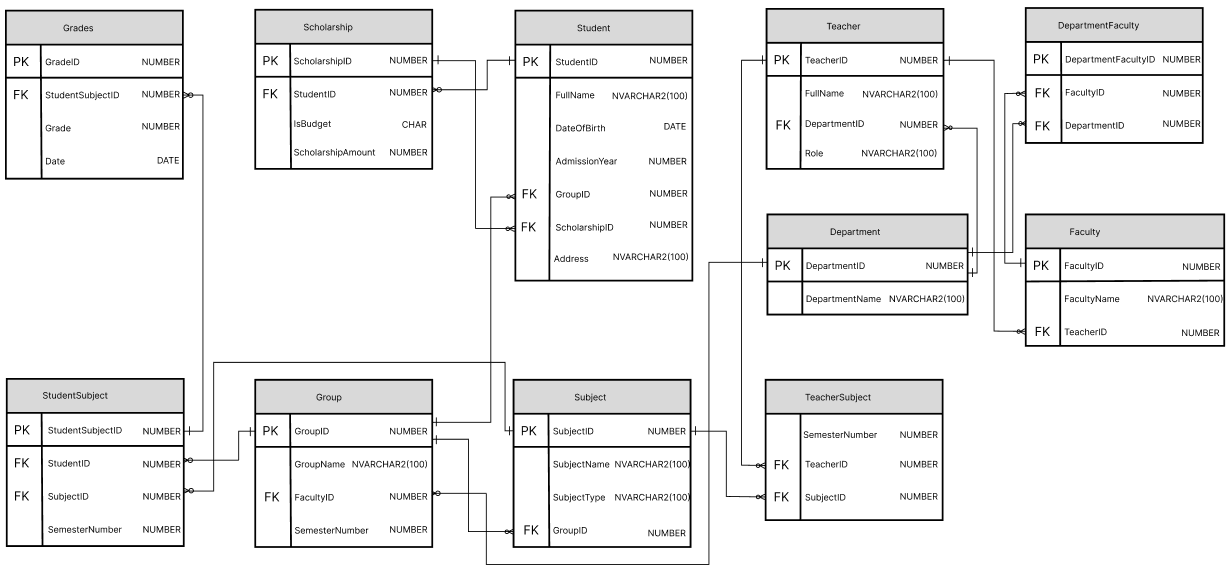


Рисунок 4.3 − Физическая схема БД для СУБД Oracle

1. Таблица Grades

* GradeID (NUMBER) — первичный ключ, уникальный идентификатор оценки.
* StudentSubjectID (NUMBER) — внешний ключ, ссылается на связь студента с предметом в таблице StudentSubject.
* Grade (NUMBER) — значение оценки, полученной студентом.
* Date (DATE) — дата, когда была выставлена оценка.

2. Таблица Scholarship

* ScholarshipID (NUMBER) — первичный ключ, уникальный идентификатор стипендии.
* StudentID (NUMBER) — внешний ключ, ссылается на таблицу Student.
* IsBudget (CHAR) — поле, определяющее, является ли студент бюджетным (государственным) или платным.
* ScholarshipAmount (NUMBER) — сумма назначенной стипендии.

3. Таблица Student

* StudentID (NUMBER) — первичный ключ, уникальный идентификатор студента.
* FullName (NVARCHAR2(100)) — полное имя студента.
* DateOfBirth (DATE) — дата рождения студента.
* AdmissionYear (NUMBER) — год поступления в университет.
* GroupID (NUMBER) — внешний ключ, ссылается на таблицу Group.
* ScholarshipID (NUMBER) — внешний ключ, ссылается на таблицу Scholarship.
* Address (NVARCHAR2(100)) — адрес проживания студента.

4. Таблица Teacher

* TeacherID (NUMBER) — первичный ключ, уникальный идентификатор преподавателя.
* FullName (NVARCHAR2(100)) — полное имя преподавателя.
* DepartmentID (NUMBER) — внешний ключ, ссылается на таблицу Department.
* Role (NVARCHAR2(100)) — роль преподавателя (например, профессор, доцент и т.д.).

5. Таблица DepartmentFaculty

* DepartmentFacultyID (NUMBER) — первичный ключ, уникальный идентификатор связи между кафедрой и факультетом.
* FacultyID (NUMBER) — внешний ключ, ссылается на таблицу Faculty.
* DepartmentID (NUMBER) — внешний ключ, ссылается на таблицу Department.

6. Таблица Department

* DepartmentID (NUMBER) — первичный ключ, уникальный идентификатор кафедры.
* DepartmentName (NVARCHAR2(100)) — название кафедры.

7. Таблица Faculty

* FacultyID (NUMBER) — первичный ключ, уникальный идентификатор факультета.
* FacultyName (NVARCHAR2(100)) — название факультета.
* TeacherID (NUMBER) — внешний ключ, ссылается на таблицу Teacher (предполагается, что это декан факультета или руководитель).

8. Таблица StudentSubject

* StudentSubjectID (NUMBER) — первичный ключ, уникальный идентификатор связи студента с предметом.
* StudentID (NUMBER) — внешний ключ, ссылается на таблицу Student.
* SubjectID (NUMBER) — внешний ключ, ссылается на таблицу Subject.
* SemesterNumber (NUMBER) — номер семестра, в котором студент изучает данный предмет.

9. Таблица Group

* GroupID (NUMBER) — первичный ключ, уникальный идентификатор группы.
* GroupName (NVARCHAR2(100)) — название группы.
* FacultyID (NUMBER) — внешний ключ, ссылается на таблицу Faculty.
* SemesterNumber (NUMBER) — номер текущего семестра для данной группы.

10. Таблица Subject

* SubjectID (NUMBER) — первичный ключ, уникальный идентификатор предмета.
* SubjectName (NVARCHAR2(100)) — название предмета.
* SubjectType (NVARCHAR2(100)) — тип предмета (например, лекция, практика и т.д.).
* GroupID (NUMBER) — внешний ключ, ссылается на таблицу Group.

11. Таблица TeacherSubject

* TeacherSubjectID (NUMBER) — первичный ключ, уникальный идентификатор связи преподавателя с предметом.
* TeacherID (NUMBER) — внешний ключ, ссылается на таблицу Teacher.
* SubjectID (NUMBER) — внешний ключ, ссылается на таблицу Subject.
* SemesterNumber (NUMBER) — номер семестра, в котором преподаватель ведет данный предмет.

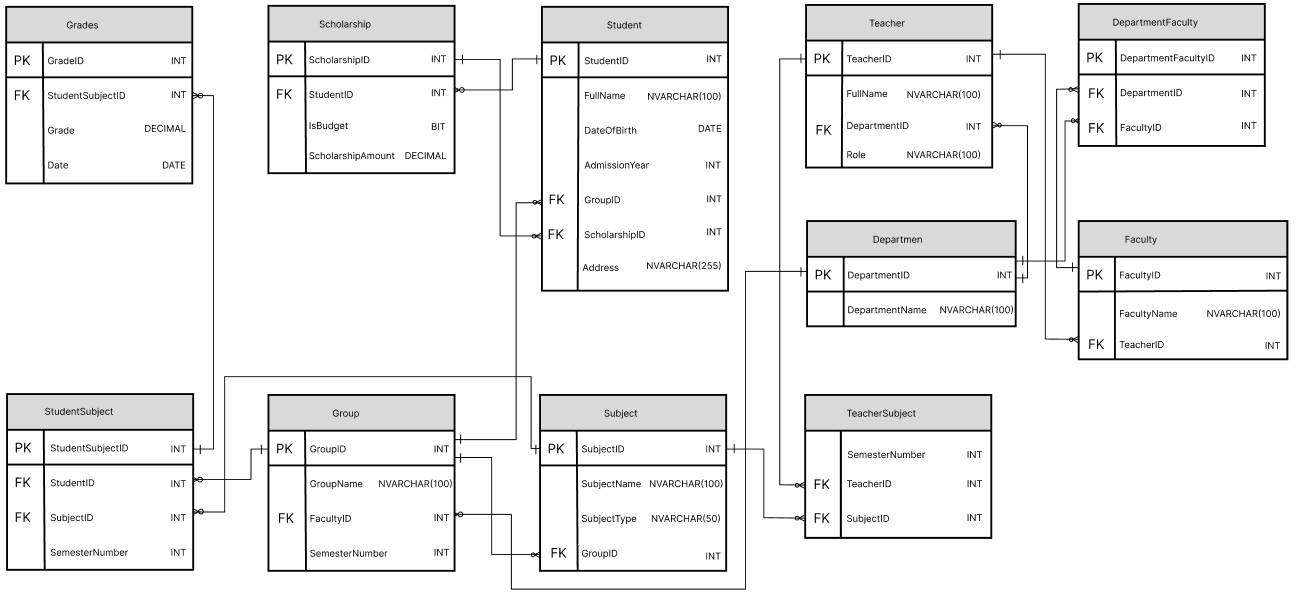


Рисунок 4.4 − Физическая схема БД для СУБД SQL Server

1. Grades (Оценки)

* GradeID (PK) - Уникальный идентификатор оценки.
* StudentSubjectID (FK) - Ссылка на таблицу StudentSubject, указывающая на предмет, по которому выставлена оценка.
* Grade - Значение оценки в десятичном формате.
* Date - Дата выставления оценки.

1. Scholarship (Стипендия)

* ScholarshipID (PK) - Уникальный идентификатор стипендии.
* StudentID (FK) - Ссылка на таблицу Student, указывающая на студента, получающего стипендию.
* IsBudget - Флаг (да/нет), указывающий, бюджетная ли это форма обучения.
* ScholarshipAmount - Сумма стипендии.

1. Student (Студент)

* StudentID (PK) - Уникальный идентификатор студента.
* FullName - Полное имя студента.
* DateOfBirth - Дата рождения студента.
* AdmissionYear - Год поступления.
* GroupID (FK) - Ссылка на таблицу Group, указывающая на группу, к которой принадлежит студент.
* ScholarshipID (FK) - Ссылка на таблицу Scholarship, указывающая на стипендию студента.
* Address - Адрес проживания студента.

1. Teacher (Преподаватель)

* TeacherID (PK) - Уникальный идентификатор преподавателя.
* FullName - Полное имя преподавателя.
* DepartmentID (FK) - Ссылка на таблицу Department, указывающая на кафедру, к которой относится преподаватель.
* Role - Роль преподавателя (например, профессор, ассистент и т.д.).

1. DepartmentFaculty (Кафедра-Факультет)

* DepartmentFacultyID (PK) - Уникальный идентификатор записи.
* DepartmentID (FK) - Ссылка на таблицу Department, указывающая на кафедру.
* FacultyID (FK) - Ссылка на таблицу Faculty, указывающая на факультет.

1. Department (Кафедра)

* DepartmentID (PK) - Уникальный идентификатор кафедры.
* DepartmentName - Название кафедры.

1. Faculty (Факультет)

* FacultyID (PK) - Уникальный идентификатор факультета.
* FacultyName - Название факультета.
* TeacherID (FK) - Ссылка на таблицу Teacher, указывающая на руководителя факультета.

1. StudentSubject (Предмет студента)

* StudentSubjectID (PK) - Уникальный идентификатор записи.
* StudentID (FK) - Ссылка на таблицу Student, указывающая на студента.
* SubjectID (FK) - Ссылка на таблицу Subject, указывающая на предмет.
* SemesterNumber - Номер семестра.

1. Group (Группа)

* GroupID (PK) - Уникальный идентификатор группы.
* GroupName - Название группы.
* FacultyID (FK) - Ссылка на таблицу Faculty, указывающая на факультет.
* SemesterNumber - Номер семестра.

1. Subject (Предмет)

* SubjectID (PK) - Уникальный идентификатор предмета.
* SubjectName - Название предмета.
* SubjectType - Тип предмета (например, лекция, семинар и т.д.).
* GroupID (FK) - Ссылка на таблицу Group, указывающая на группу.

1. TeacherSubject (Предмет преподавателя)

* TeacherSubjectID (PK) - Уникальный идентификатор записи.
* TeacherID (FK) - Ссылка на таблицу Teacher, указывающая на преподавателя.
* SubjectID (FK) - Ссылка на таблицу Subject, указывающая на предмет.
* SemesterNumber - Номер семестра.