**Part 1: Выбор Сценария**

Для данной работы выбран сценарий: Система высшего образования: Студенты, предметы, преподаватели, занятие и т. д.

**Part 2: Проектирование Базы Данных и Документация**

Идентификация Сущностей и Атрибутов:

1. Студенты (Students):

2. Предметы (Subjects):

3. Преподаватели (Professors):

4. Занятие (Class):

**Проектирование Таблиц:**

1. **Table Name: Студент (Students):**

**Description:** Хранит информацию о студентах.

**Attributes:**

• StudentID (INT, PK)

• FirstName (VARCHAR(100))

• LastName (VARCHAR(100))

• Email (VARCHAR(100), UNIQUE)

• PhoneNumber (VARCHAR(20))

1. **Table Name:**  **Предмет (Subjects):**

**Description:** Хранит информацию о предметах.

**Attributes:**

• SubjectID (INT, PK)

• SubjectName (VARCHAR(100))

• Description (TEXT)

• NumberSubject (INT, CHECK (NumberSubject > 0))

**3. Table Name: Преподаватель (Professors):**

**Description:** Хранит информацию о преподавателях.

**Attributes:**

• ProfessorID (INT, PK)

• FirstName (VARCHAR(100))

• LastName (VARCHAR(100))

• Email (VARCHAR(100), UNIQUE)

1. **Table Name: Занятие (Classes):**

**Description:** Хранит информацию о занятиях.

**Attributes:**

• ClassID (INT, PK)

• SubjectID (INT, FK, NOT NULL)

• ProfessorID (INT, FK, NOT NULL)

• StartTime (TIME)

• Room (VARCHAR(100))

1. **Table Name: Запись на курсы (Enrollments):**

**Description:** Хранит информацию о записи студентов на курсы.

**Attributes:**

• EnrollmentID (INT, PK)

• StudentID (INT, FK, NOT NULL)

• ClassID (INT, FK)

• EnrollmentDate (DATE)

Все таблицы, соответствуют третьей нормальной форме (3NF).

**Students:** Все атрибуты атомарны и зависят только от StudentID (PK).

**Subjects**: Все атрибуты атомарны и зависят только от SubjectID (PK).

**Professors:** Все атрибуты атомарны и зависят только от ProfessorID (PK).

**Classes:** Все атрибуты атомарны и зависят только от ClassID (PK). SubjectID и ProfessorID - внешние ключи, обеспечивающие связь с другими таблицами.

**Enrollments:** Все атрибуты атомарны и зависят только от EnrollmentID (PK). StudentID и ClassID - внешние ключи.

Таким образом, все условия 3NF соблюдены: атомарность, зависимость от PK и отсутствие транзитивной зависимости.

**Взаимосвязи:**

**Students - Enrollments: Один-ко-многим:** Один студент может быть записан на много занятий, но каждая запись на курс относится к одному конкретному студенту.

(Enrollments.StudentID является внешним ключом, ссылающимся на Students.StudentID)

• **Classes - Enrollments: Один-ко-многим:** На одно занятие может быть записано много студентов, но каждая запись относится только к одному конкретному занятию.

(Enrollments.ClassID является внешним ключом, ссылающимся на Classes.ClassID)

**• Subjects - Classes: Один-ко-многим:** У одного предмета может быть много занятий, но на каждом занятие изучается один предмет.

(Classes.SubjectID является внешним ключом, ссылающимся на

Subjects.SubjectID)

**• Professors - Classes: Один-ко-многим**: Один преподаватель может вести много занятий, но на каждом занятии может быть только один преподаватель. (Classes.ProfessorID является внешним ключом, ссылающимся на Professors.ProfessorID)

Связь **Многие-ко-Многим между** Студентами (Students) и Предметами (Subjects) реализована через таблицу Запись на курсы (Enrollments).

**Студент может быть записан на несколько курсов:** Для каждого курса, на который записан студент, будет создана отдельная запись в таблице Enrollments. StudentID будет указывать на конкретного студента, а ClassID - на конкретное занятие, относящееся к предмету.

**Курс (через Занятие) может быть посещен многими студентами:** Для каждого студента, посещающего занятие по этому курсу, будет создана отдельная запись в таблице Enrollments. ClassID будет указывать на конкретное занятие, а StudentID - на конкретного студента.

Таким образом, наличие таблицы Enrollments, связывающей Students и Classes (а через Classes - и Subjects), позволяет реализовать связь **"Многие-ко-Многим"** между студентами и предметами.

**Part 3: ER-Диаграмма**

