МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Механико-математический факультет

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «РАСПРЕДЕЛЕНИЕ УЧЕБНОЙ НАГРУЗКИ»

Выполнил:

Студент 2 курса, 9 группы

Куркин Ян Александрович

Минск, 2023

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ 1

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ 1

Механико-математический факультет 1

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «РАСПРЕДЕЛЕНИЕ УЧЕБНОЙ НАГРУЗКИ» 1

1. Постановка задачи 3

2. Инициирование проекта (фаза 0) 4

3. Определение множеств сущностей (фаза 1) 5

3.1. Выделение множеств сущностей 5

3.2. Описание множеств сущностей 5

4. Построение модели уровня сущностей (фаза 2) 8

4.1. Матрица связей 8

4.2. Описание связей 8

5. Построение модели уровня ключей. 10

5.1. Разрешение неопределенных связей 10

5.2. Определение ключевых атрибутов и доменов 10

5.3. Концептуальная схема 13

6. Построение полноатрибутной модели (фаза 4). 13

7. Проведение нормализации отношений. 16

7.1. Постановка проблемы. 16

7.2. Результат проведения нормализации. 17

8. Построение даталогической модели 24

Предмет 24

Группа 24

Преподаватель 25

Расписание занятий 26

Занятие 27

Нагрузка занятий 27

1. Постановка задачи

При распределении учебной нагрузки появляется необходимость в ведении системы для её учёта. Преподаватель определённой кафедры должен обеспечивать проведение занятий по некоторым предметам определённое количество часов. Система должна удовлетворять следующим требованиям:

* Каждый предмет должен быть назначен хотя бы одному преподавателю для определенной группы студентов.
* Общее количество часов, распределенных среди преподавателей по каждому предмету, должно соответствовать общему количеству часов, указанному для этого предмета в таблице "Предметы".
* Преподаватели с более высокими учеными степенями и стажем работы должны иметь приоритет при распределении сложных и специализированных предметов.
* Распределение учебной нагрузки между преподавателями должно быть сбалансировано, чтобы избежать перегрузки некоторых преподавателей, в то время как другие имеют меньшую нагрузку. Это также должно учитывать административные обязанности преподавателей, занимающих руководящие должности.
* Нагрузка должна быть равномерно распределена между группами студентов, чтобы избежать перегрузки одной группы по сравнению с другими.

Команда, работавшая над проектом:

1. Михалевич Владислав. Построение концептуальной схемы уровня сущностей. Составление описание сущностей. Участие в создании полноатрибутной модели.
2. Куркин Ян. Построение концептуальной схемы уровня ключей и полноатрибутной схемы. Описание связей.
3. Кленицкая София. Заполнение базы данных. Составление описания сущностей. Примеры сущностей

2. Инициирование проекта (фаза 0)

Допустим, что на этой фазе были определены все участники проекта, определены их функции. Уточним цели проектирования. В результате предварительного изучения поставленной задачи определены требования к проектируемой системе:

* Необходимо отразить в системе распределение нагрузки для преподавателей.
* Группы определяются исходя из курса и её номера. Некоторые группы могут и отсутствовать
* У одной группы в одно и то же время не может быть более одного занятия
* Раз в учебный семестр или в случае, если преподаватель сменился во время обучения, за ним закрепляются группы и предметы.
* Каждая учебная группа должна иметь не слишком большое число студентов
* В разных группах по одному и тому же предмету могут вести занятия разные преподаватели, так и один преподаватель может вести по разным предметам в разных классах.

Сформулированные требования определяют основные исходные данные для остальных фаз проектирования системы.

3. Определение множеств сущностей (фаза 1)

3.1. Выделение множеств сущностей

В соответствии с описанием предметной области можно выделить следующие множества сущностей:

* Предмет
* Преподаватель
* Группа
* Занятие
* Расписание

3.2. Описание множеств сущностей

Описание выделенных множеств сущностей приведено в табл. 1.

Табл. 1. Пул сущностей

| Имя множества сущностей | Определение множества сущностей | Описание множества сущностей |
| --- | --- | --- |
| Предмет | Учебная дисциплина, по которой проводятся занятия в университете | Дисциплина определяется указаниями вышестоящей организации или руководством университета. Как только соответствующая организация или руководство университета принимают решение о введении новой учебной дисциплины, формируется новый экземпляр данного множества сущностей |
| Преподаватель | Физическое лицо, имеющее высшее образование | Как только конкретный человек устраивается на работу в данный университет, формируется новый экземпляр данного множества сущностей |
| Группа | Совокупность студентов, являющиеся «структурной единицей», для которой будут проводиться занятия | Формируется в начале учебного года. Когда абитуриенты зачисляются в университет и причисляются к созданному экземпляру сущности. В случае, если количество студентов стало слишком мало, группа может быть расформирована и экземпляр будет удален. Если студент исключается из группы, количество студентов в группе изменяется. |
| Занятие | Сущность, связывающая между собой предмет, преподавателя и группу. Содержит информацию о времени занятия и аудитории. | Формируется в начале каждого семестра. |
| Расписание | Сущность, хранящая количество академических часов для каждого занятия. | Формируется в начале каждого семестра. |

Приведем примеры сущностей:

* Предмет: Математический анализ, технологии программирования
* Преподаватель: Козлов Дмитрий Сергеевич, преподает математический анализ, Огурцова Марья Викторовна, преподает технологии программирования
* Группа: 2 курс, 9Б группа
* Занятие: Математический анализ, 2 курс, 9 группа, Козлов Дмитрий Сергеевич, 15/02/2023 11:15:00, 609.
* Расписание: занятие, которое было выше, 30 академических часов.

4. Построение модели уровня сущностей (фаза 2)

4.1. Матрица связей

На основании анализа постановки задачи устанавливаются связи между множествами сущностей. Матрица связей приведена в табл. 2.

Таблица 2. Матрица связей.

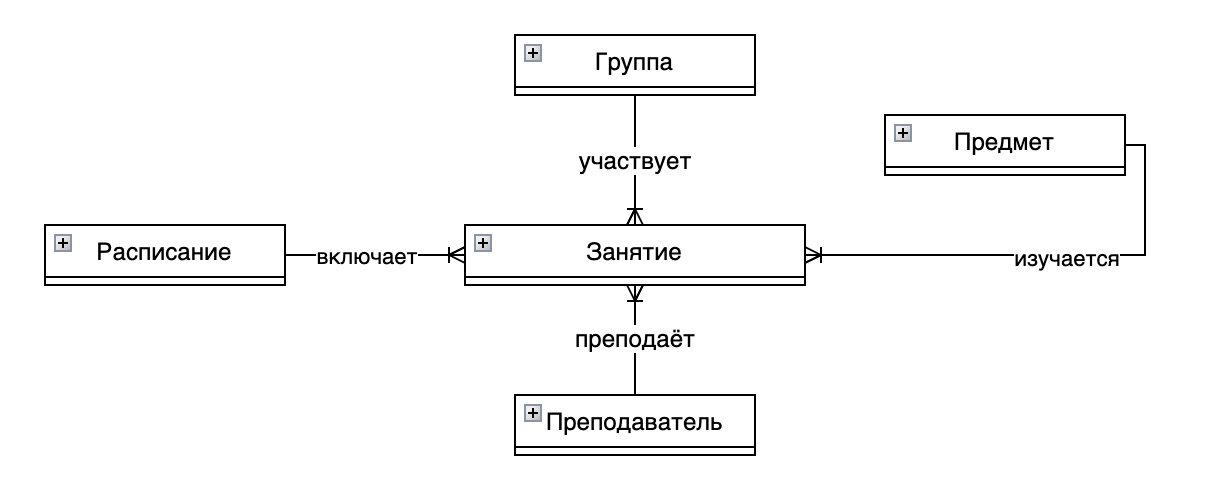
|  | Предмет | Преподаватель | Группа | Занятие | Расписание |
| --- | --- | --- | --- | --- | --- |
| Предмет |  |  |  | X |  |
| Преподаватель |  |  |  | X |  |
| Группа |  |  |  | X |  |
| Занятие | X | X | X |  | X |
| Расписание |  |  |  | X |  |

4.2. Описание связей

Таблица 3. Описание связей

| 1-я сущность | 2-я сущность | Имя связи | Тип связи | Мощность связи | Описание связи |
| --- | --- | --- | --- | --- | --- |
| Предмет | Занятие | изучается | идент. опред. | 1: M | На одном занятии может изучаться один предмет. Один предмет может изучаться на занятиях |
| Преподаватель | Занятие | преподаёт | идент. опред. | 1: M | Одно занятие может проводиться только одним преподавателем. Преподаватель может проводить одно или несколько занятий |
| Группа | Занятие | участвует | идент. опред. | 1: M | Одна и несколько групп могут быть на одном занятии. Одна группа может быть только на одном занятии |
| Расписание | Занятие | включает | идент. опред. | 1: M | Расписание включает в себя одно и более занятий. |

Рис. 1. Концептуальная схема уровня сущностей

5. Построение модели уровня ключей.

5.1. Разрешение неопределенных связей

Стоит отметить, что неопределенных связей (типа «многие-ко-многим) в рассматриваемом примере нет. Следовательно необходимости в разрешении таковых связей отсутствует.

5.2. Определение ключевых атрибутов и доменов

Определение ключевых атрибутов осуществляется одновременно с определением доменов, на которых определяются эти атрибуты. Сначала из анализа задачи для каждого множества сущностей выделяются ключи(простые или составные), позволяющие идентифицировать экземпляры сущностей. В случае если ключей несколько, один выбирается в качестве первичного ключа, а остальные определяются как альтернативные. Связи между множествами сущностей представляются с помощью внешних ключей. Таким образом, определяются все ключи. Для каждого атрибута определяется домен.

В рассматриваемом примере для выделенных сущностей можно предложить следующие ключи и домены:

**Группа**

Первичный ключ: ID\_группы. Значениями атрибута являются натуральные числа (порядковый номер в перечислении предметов), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

**Предмет**

Первичный ключ: ID\_предмета. Значениями атрибута являются натуральные числа (порядковый номер в перечислении предметов), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

Альтернативный ключ: Название предмета. Значения предоставлены типом nvarchar(50)

**Расписание**

Первичный ключ: ID\_расписания. Значениями атрибута являются натуральные числа (порядковый номер в перечислении предметов), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

Внешний ключ: Так как множество сущностей «расписание» связано с множеством сущностей «занятие», создаётся внешний ключ: ID\_занятия, представленный в виде типа данных int.

**Преподаватель**

Первичный ключ: ID\_преподавателя. Значениями атрибута являются натуральные числа (порядковый номер в перечислении предметов), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

**Занятие**

Первичный ключ: ID\_занятия. Значениями атрибута являются натуральные числа (порядковый номер в перечислении предметов), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

Внешний ключ: Так как множество сущностей «занятие» связано с множеством сущностей «предмет», создаётся внешний ключ: ID\_предмета, представленный в виде типа данных int.

Внешний ключ: Так как множество сущностей «преподаватель» связано с множеством сущностей «предмет», создаётся внешний ключ: ID\_преподавателя, представленный в виде типа данных int.

Внешний ключ: Так как множество сущностей «группа» связано с множеством сущностей «предмет», создаётся внешний ключ: ID\_группы, представленный в виде типа данных int.

Описание ключевых атрибутов приведено в таблице 4.

Таблица 4. Описание доменов

| Множество сущностей | Имя атрибута | Тип данных | Признак обязательности | Примечание |
| --- | --- | --- | --- | --- |
| Группа | Код группы | int | NOT NULL | PK |
| Предмет | Код предмета | int | NOT NULL | PK |
| Название предмета | nvarchar(50) | NOT NULL | AK |
| Преподаватель | Код преподавателя | int | NOT NULL | PK |
| Расписание | Код расписания | int | NOT NULL | PK |
| Код занятия | Int | NOT NULL | FK |
| Занятие | Код занятия | Int | NOT NULL | PK |
| Код группы | Int | NOT NULL | FK1 |
| Код предмета | Int | NOT NULL | FK2 |
| Код преподавателя | Int | NOT NULL | FK3 |

5.3. Концептуальная схема

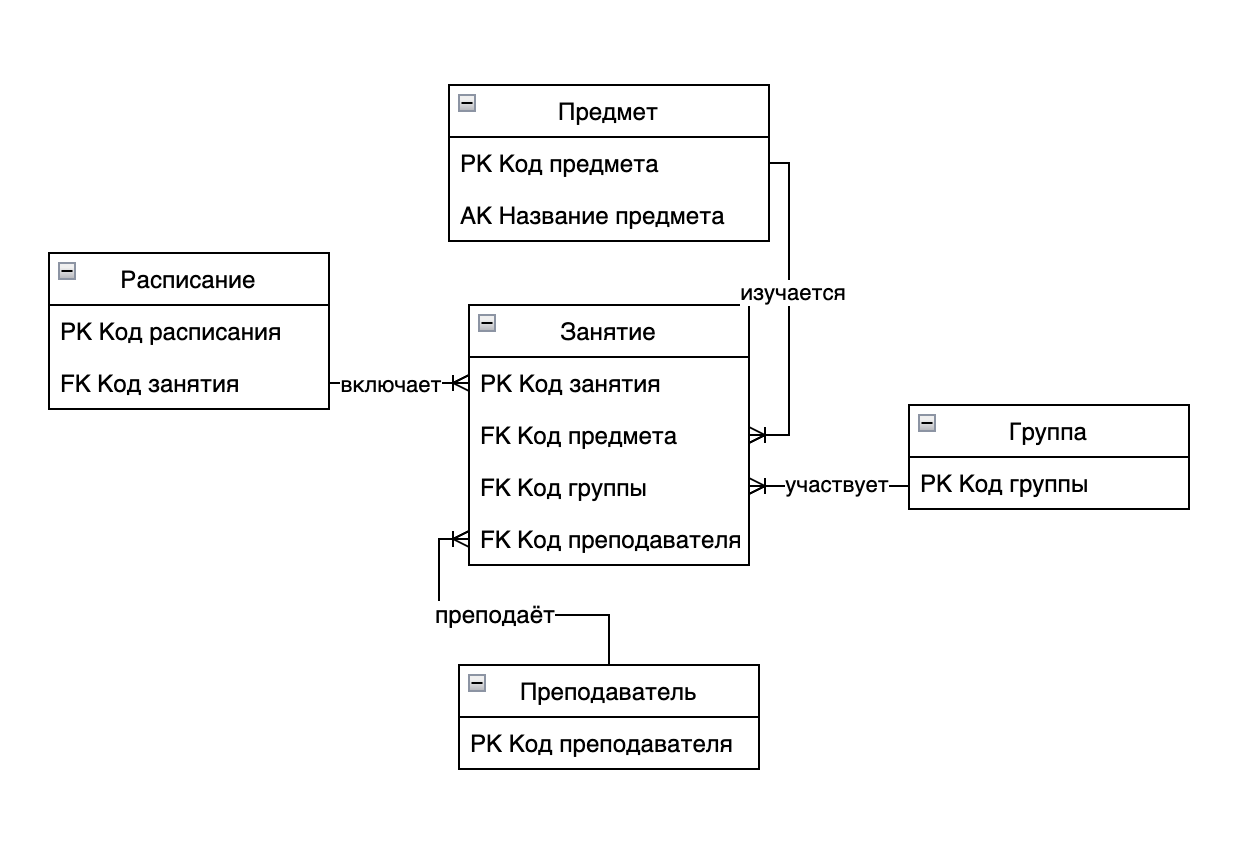
По результатам определения первичным ключей строится концептуальная схема уровня ключей (рис. 2).

Рис. 2. Концептуальная схема уровня ключей

6. Построение полноатрибутной модели (фаза 4).

На этом этапе в соответствии с проведенным анализом предметной области определяются все необходимые атрибуты и домены, на которых эти атрибуты определены. Соответствующим образом модифицируются таблица описания доменов и атрибутов (табл. 5) и строится полноатрибутная концептуальная схема (рис. 3).

Таблица 5. Описание атрибутов

| Множество сущностей | Имя атрибута | Тип данных | Признак обязательности | Примечание |
| --- | --- | --- | --- | --- |
| Преподаватель | Код преподавателя | int | NOT NULL | PK |
| Фамилия | nvarchar(30) | NOT NULL |  |
| Имя | nvarchar(30) | NOT NULL |  |
| Отчество | nvarchar(30) | NULL |  |
| Адрес | nvarchar(30) | NOT NULL |  |
| Телефон | nvarchar(30) | NOT NULL |  |
| Ученая степень | nvarchar(30) | NOT NULL |  |
| Административная должность | nvarchar(30) | NOT NULL |  |
| Стаж работы | tinyint | NOT NULL |  |
| Предмет | Код предмета | int | NOT NULL | PK |
| Название предмета | nvarchar(30) | NOT NULL | AK |
| Расписание | Код расписания | int | NOT NULL | PK |
| Код занятия | int | NOT NULL | FK |
| Количество академических часов | Int | NOT NULL | Является целым положительным числом. |
| Группа | Код группы | Int | NOT NULL | PK |
| Курс | tinyint | NOT NULL | Значение в промежутке от 1 до 5. По умолчанию: 1. |
| Номер | nvarchar(10) | NOT NULL | Номер группы должен быть уникальным |
| Количество студентов | int | NOT NULL | Значение находится в промежутке от 5 до 30 |
| Занятие | Код занятия | int | NOT NULL | PK |
| Код предмета | Int | NOT NULL | FK1 |
| Код группы | Int | NOT NULL | FK2 |
| Код преподавателя | int | NOT NULL | FK3 |
| Время занятия | Time | NOT NULL | По умолчанию:  08:15:00 |
| Аудитория | nvarchar(50) | NOT NULL |  |

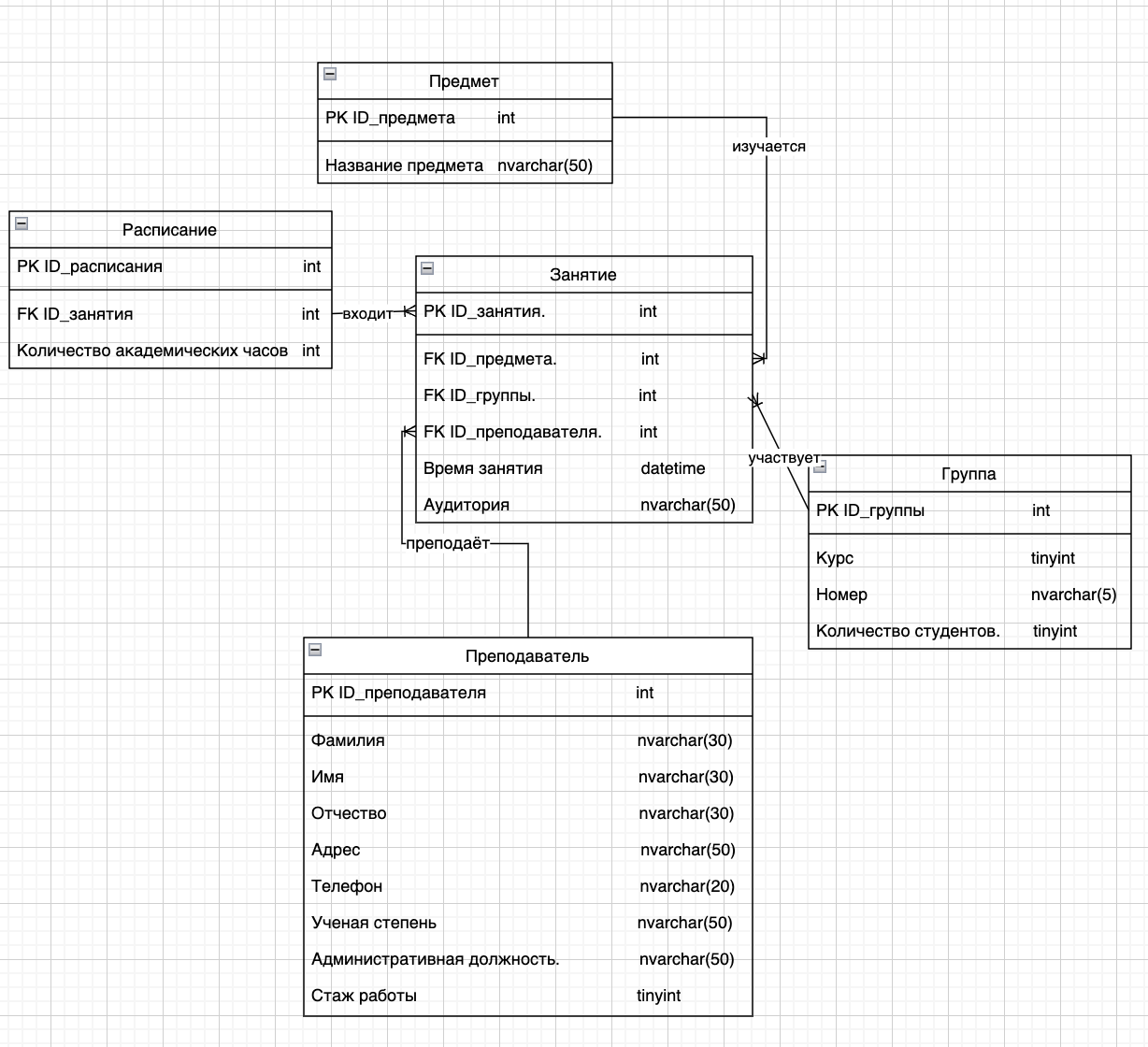


Рис 3. Полноатрибутная схема.

7. Проведение нормализации отношений.

7.1. Постановка проблемы.

Из анализа поставленной задачи можно установить следующие неточности, допущенные при проектировании: таблица расписания будет содержать избыточную информацию с содержащимися дубликатами. Так как может меняться время проведения занятия и аудитория, в которой оно будет проводится. Может возникнуть ситуация, что казалось бы для занятия, которое проводится одним преподавателем для одной и той же группы по одному тому же предмету будет разное количество академических часов.

Для решения возникшей проблемы можно провести нормализацию базы данных и вынести информацию о времени занятия и аудитории, в которой оно проводится в отдельную сущность, которую мы назовём «расписание занятий». Так же будет переименована сущность «расписание» в «нагрузка занятий».

7.2. Результат проведения нормализации.

Описание добавленных сущностей:

Таблица 6. Добавленные сущности.

| Имя сущности | Определение сущности | Описание сущности |
| --- | --- | --- |
| Расписание занятий | Содержит информацию о времени проведения занятия и аудитории, в которой оно проводится. | Содержит информацию о времени проведения занятия и аудитории, в которой оно проводится. |
| Нагрузка занятий | Содержит информацию о количестве академических часов, выделенных на данное занятие | Содержит информацию о количестве академических часов, выделенных на данное занятие |

Как результат проведения нормализации были сделано следующее:

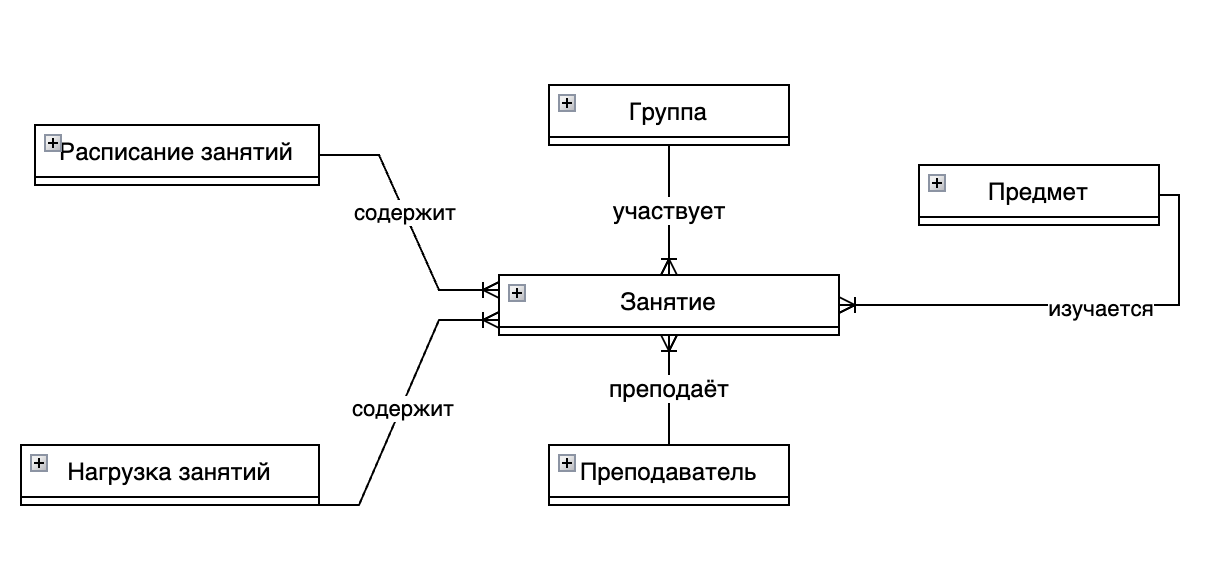
1. Обновленная таблица связей.
2. Обновленное описание связей.
3. Обновленная концептуальная схема уровня сущностей.
4. Обновленная концептуальная схема уровня ключей.
5. Обновленная таблица описания доменов и атрибутов.
6. Обновленная полноатрибутная концептуальная схема.

Таблица 7. Обновленная матрица связей.

|  | Группа | Предмет | Преподаватель | Занятие | Расписание занятий | Нагрузка занятий |
| --- | --- | --- | --- | --- | --- | --- |
| Группа |  |  |  | X |  |  |
| Предмет |  |  |  | X |  |  |
| Преподаватель |  |  |  | X |  |  |
| Занятие |  |  |  |  | X | X |
| Расписание занятий |  |  |  | X |  |  |
| Нагрузка занятий |  |  |  | X |  |  |

Таблица 8. Обновленное описание связей.

| 1-я сущность | 2-я сущность | Имя связи | Тип связи | Мощность связи | Описание связи |
| --- | --- | --- | --- | --- | --- |
| Предмет | Занятие | изучается | идент. опред. | 1: M | На одном занятии может изучаться один предмет. Один предмет может изучаться на занятиях |
| Преподаватель | Занятие | преподаёт | идент. опред. | 1: M | Одно занятие может проводиться только одним преподавателем. Преподаватель может проводить одно или несколько занятий |
| Группа | Занятие | участвует | идент. опред. | 1: M | Одна и несколько групп могут быть на одном занятии. Одна группа может быть только на одном занятии |
| Расписание занятий | Занятие | Содержит | идент. опред. | 1: M | Несколько занятий может находиться в одном расписании занятий. |
| Нагрузка занятий | Занятие | Содержит | идент. опред. | 1: M | Несколько занятий содержатся только в одном нагрузке занятий. |

Рис 4. Обновленная концептуальная схема уровня сущностей.

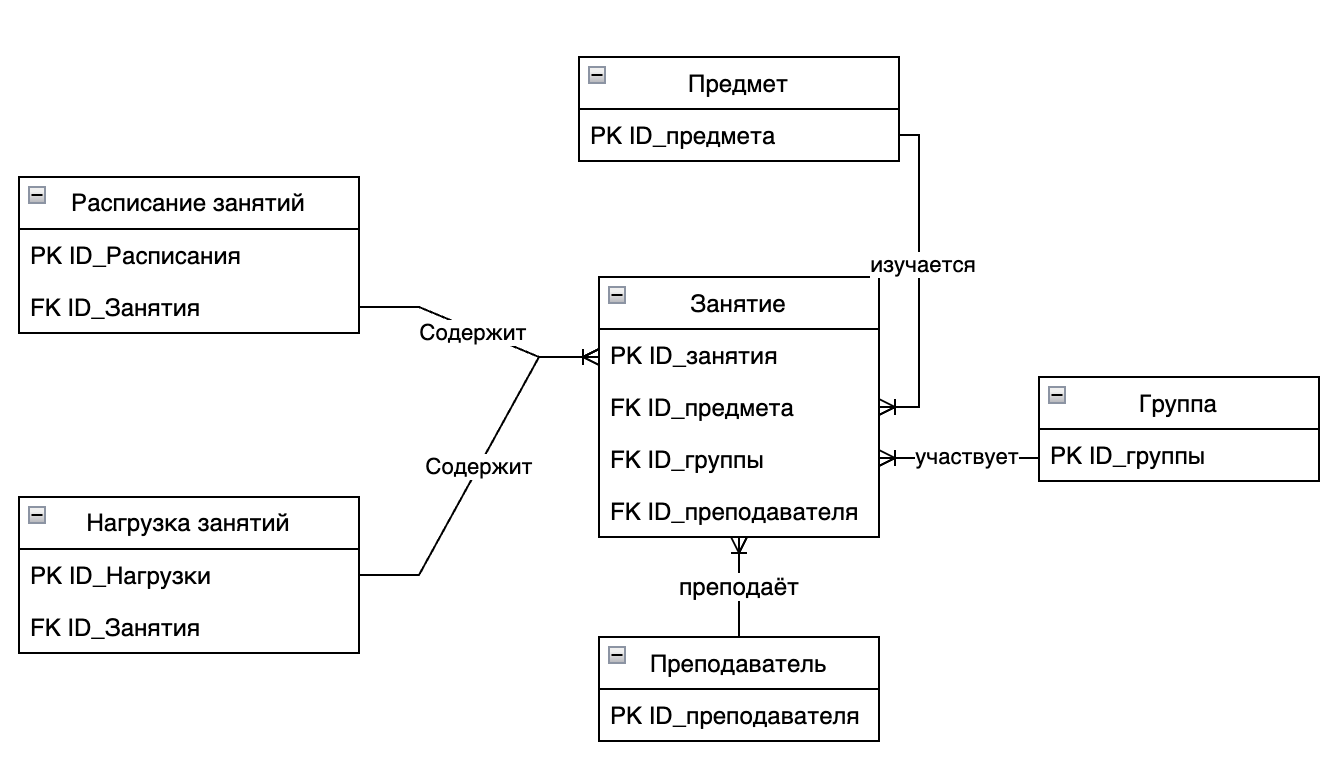


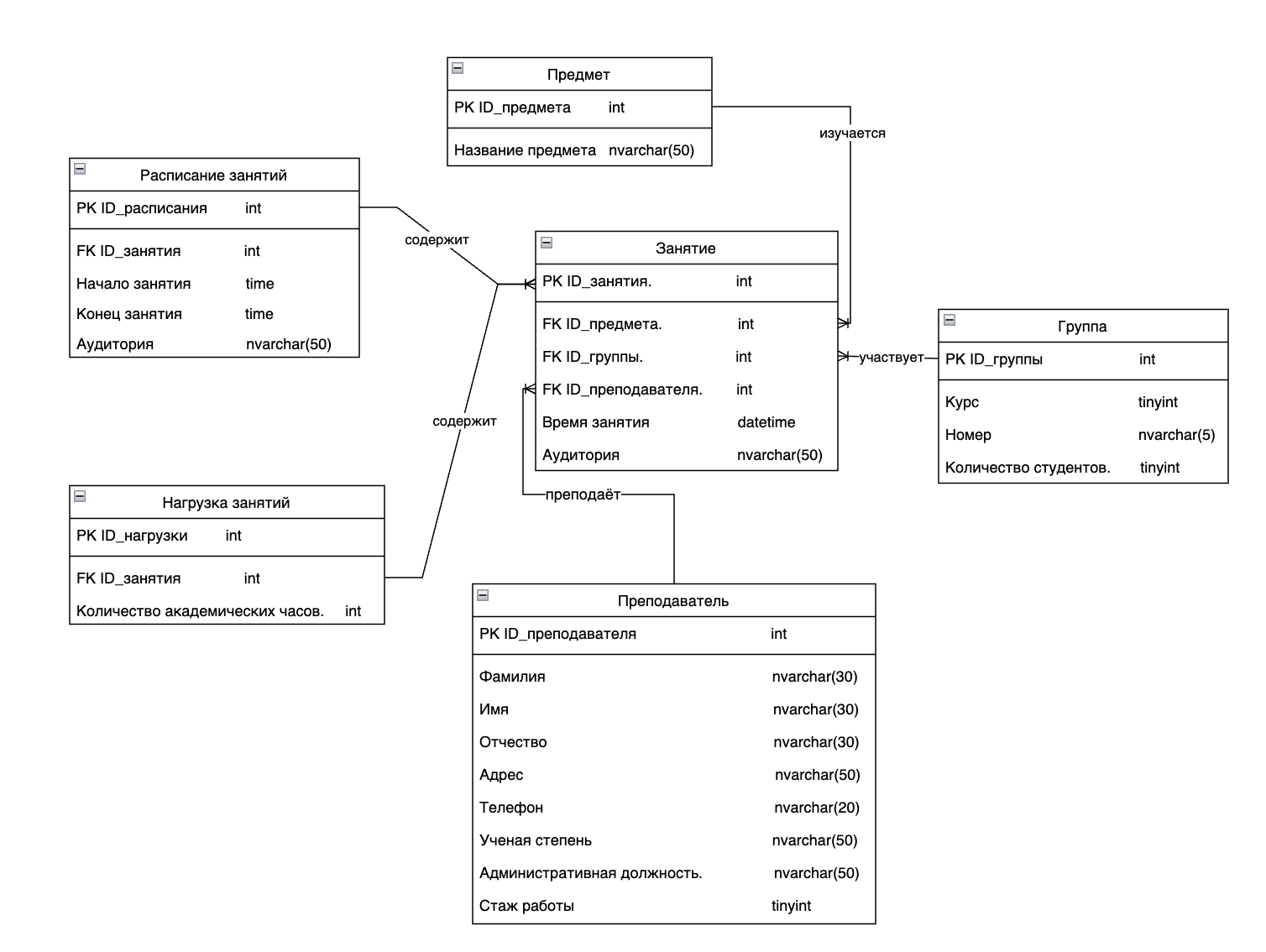
Рис 5. Обновленная концептуальная схема уровня ключей.

| Множество сущностей | Имя атрибута | Тип данных | Признак обязательности | Примечание |
| --- | --- | --- | --- | --- |
| Группа | ID\_группы | int | NOT NULL | PK |
| Предмет | ID предмета | int | NOT NULL | PK |
| Название предмета | nvarchar(50) | NOT NULL | AK |
| Преподаватель | ID преподавателя | int | NOT NULL | PK |
| Расписание  Занятий | ID расписания | int | NOT NULL | PK |
| ID занятия | int | NOT NULL | FK |
| Занятие | ID занятия | int | NOT NULL | PK |
| ID группы | int | NOT NULL | FK1 |
| ID предмета | int | NOT NULL | FK2 |
| ID преподавателя | int | NOT NULL | FK3 |
| Нагрузка занятий | ID нагрузки | int | NOT NULL | PK |
| ID занятия | Int | NOT NULL | FK |

Таблица. 9. Обновленное описание доменов.

Таблица 10. Обновленное описание атрибутов

| Множество сущностей | Имя атрибута | Тип данных | Признак обязательности | Примечание |
| --- | --- | --- | --- | --- |
| Преподаватель | Код преподавателя | int | NOT NULL | PK |
| Фамилия | nvarchar(30) | NOT NULL |  |
| Имя | nvarchar(30) | NOT NULL |  |
| Отчество | nvarchar(30) | NULL |  |
| Адрес | nvarchar(30) | NOT NULL |  |
| Телефон | nvarchar(30) | NOT NULL |  |
| Ученая степень | nvarchar(30) | NOT NULL |  |
| Административная должность | nvarchar(30) | NOT NULL |  |
| Стаж работы | tinyint | NOT NULL |  |
| Предмет | Код предмета | int | NOT NULL | PK |
| Название предмета | nvarchar(30) | NOT NULL | AK |
| Расписание | Код расписания | int | NOT NULL | PK |
| Код занятия | int | NOT NULL | FK |
| Начало занятия | time | NOT NULL |  |
| Конец занятия | time | NOT NULL | Конец занятия не должен быть раньше, чем начало этого самого занятия |
| Группа | Код группы | Int | NOT NULL | PK |
| Курс | tinyint | NOT NULL | Значение в промежутке от 1 до 5. По умолчанию: 1. |
| Номер | nvarchar(10) | NOT NULL | Номер группы должен быть уникальным |
| Количество студентов | int | NOT NULL | Значение находится в промежутке от 5 до 30 |
| Занятие | Код занятия | int | NOT NULL | PK |
| Код предмета | Int | NOT NULL | FK1 |
| Код группы | Int | NOT NULL | FK2 |
| Код преподавателя | int | NOT NULL | FK3 |
| Нагрузка занятий | ID нагрузки | int | NOT NULL | PK |
| ID занятия | int | NOT NULL | FK |
| Количество академических часов | int | NOT NULL | Является положительным числом |

Рис. 6. Обновленная полноатрибутная схема.

8. Построение даталогической модели

Создадим базу данных Schedule

**USE** master;

**DROP** **DATABASE** **IF** **EXISTS** Schedule;

**CREATE** **DATABASE** Schedule

**COLLATE** Cyrillic\_General\_CI\_AS;

**USE** Schedule;

В соответствии с приведенной структурой, внутренняя схема разрабатываемой базы данных представлена в таблицах 6 - 10.

Предмет

Таблица 11. Предмет / Subject

| Имя атрибута | Имя столбца | Тип данных | Признак обязательности | Ключи |
| --- | --- | --- | --- | --- |
| ID предмета | SubjectID | int | NOT NULL | PK |
| Название предмета | SubjectName | nvarchar(50) | NOT NULL | AK |

Описание таблицы на SQL:

**CREATE** **TABLE** [Subject] (

SubjectID **int** **IDENTITY** **NOT** **NULL** **PRIMARY** **KEY** **CLUSTERED**,

SubjectName **nvarchar**(50) **NOT** **NULL**,

**CONSTRAINT** AK\_Subject\_SubjectName **UNIQUE** (SubjectName)

);

Группа

Таблица 7. Группа / Group

| Имя атрибута | Имя столбца | Тип данных | Признак обязательности | Ключи |
| --- | --- | --- | --- | --- |
| ID группы | GroupID | int | NOT NULL | PK |
| Курс | Course | tinyint | NOT NULL |  |
| Номер группы | GroupNumber | nvarchar(10) | NOT NULL |  |
| Количество студентов | AmountOfStudents | tinyint | NOT NULL |  |

Описание таблицы на SQL:

**CREATE** **TABLE** [Group] (

GroupID **int** **IDENTITY** **NOT** **NULL** **PRIMARY** **KEY** **CLUSTERED**,

Course **tinyint** **NOT** **NULL** **DEFAULT** 1,

**CONSTRAINT** CK\_Group\_Course **CHECK** (Course **BETWEEN** 1 **AND** 5),

GroupNumber **nvarchar**(10) **NOT** **NULL** **UNIQUE**,

AmountOfStudents **tinyint** **NOT** **NULL** **DEFAULT** 10,

**CONSTRAINT** CK\_GROUP\_AMOUNTOFSTUDENTS **CHECK** (AmountOfStudents **BETWEEN** 10 **AND** 30),

)

Преподаватель

Таблица 12. Преподаватель / Teacher

| Имя атрибута | Имя столбца | Тип данных | Признак обязательности | Ключи |
| --- | --- | --- | --- | --- |
| ID преподавателя | TeacherID | int | NOT NULL | PK |
| Фамилия | Surname | nvarchar(30) | NOT NULL |  |
| Имя | Name | nvarchar(30) | NOT NULL |  |
| Отчество | Patronymic | nvarchar(30) | NULL |  |
| Адрес | Address | nvarchar(50) | NOT NULL |  |
| Телефон | Phone | nvarchar(20) | NOT NULL |  |
| Ученая степень | AcademicDegree | nvarchar(50) | NOT NULL |  |
| Административная должность | AcademicPosition | nvarchar(50) | NOT NULL |  |
| Стаж работы | Seniority | tinyint | NOT NULL |  |

Описание таблицы на SQL:

**CREATE** **TABLE** Teacher (

TeacherID **int** **IDENTITY** **NOT** **NULL** **PRIMARY** **KEY** **CLUSTERED**,

Surname **nvarchar**(30) **NOT** **NULL**,

Name **nvarchar**(30) **NOT** **NULL**,

Patronymic **nvarchar**(30) **NULL**,

Adress **nvarchar**(50) **NOT** **NULL**,

Phone **nvarchar**(20) **NOT** **NULL**,

AcademicDegree **nvarchar**(50) **NOT** **NULL**,

AcademicPosition **nvarchar**(50) **NOT** **NULL**,

Seniority **tinyint** **NOT** **NULL**,

)

Расписание занятий

Таблица 13. Расписание занятий / Class Schedule

| Имя атрибута | Имя столбца | Тип данных | Признак обязательности | Ключи |
| --- | --- | --- | --- | --- |
| ID расписания | ClassScheduleID | int | NOT NULL | PK |
| ID занятия | ClassID | int | NOT NULL | FK |
| Начало занятия | StartTime | Time | NOT NULL |  |
| Конец занятия | EndTime | Time | NOT NULL |  |
| Аудитория | Auditorium | nvarchar(50) | NOT NULL |  |

Описание таблицы на SQL:

**CREATE** **TABLE** ClassSchedule (

ClassScheduleID **int** **IDENTITY** **NOT** **NULL** **PRIMARY** **KEY** **CLUSTERED**,

ClassID **int**,

StartTime **time**,

EndTime **time**,

**CONSTRAINT** CK\_ClassSchedule\_EndTime **CHECK** (EndTime > StartTime),

Auditorium **nvarchar**(50)

)

Занятие

Таблица 14. Занятие / Class

| Имя атрибута | Имя столбца | Тип данных | Признак обязательности | Ключи |
| --- | --- | --- | --- | --- |
| ID занятия | ClassID | int | NOT NULL | PK |
| ID предмета | SubjectID | int | NOT NULL | FK1 |
| ID группы | GroupID | int | NOT NULL | FK2 |
| ID преподавателя | TeacherID | int | NOT NULL | FK3 |

Описание таблицы на SQL:

**CREATE** **TABLE** Class (

ClassID **int** **IDENTITY** **NOT** **NULL** **PRIMARY** **KEY** **CLUSTERED**,

ClassTypeID **int**,

GroupID **int**,

SubjectID **int**,

TeacherId **int**,

)

Нагрузка занятий

Таблица 15. Нагрузка занятий / Class Load

| Имя атрибута | Имя столбца | Тип данных | Признак обязательности | Ключи |
| --- | --- | --- | --- | --- |
| ID нагрузки | ClassPlanID | int | NOT NULL | PK |
| ID занятия | ClassID | int | NOT NULL | FK |
| Количество академических часов | AcademicHoursCount | int | NOT NULL |  |

Описание таблицы на SQL:

**CREATE** **TABLE** ClassPlan (

ClassPlanID **int** **IDENTITY** **NOT** **NULL** **PRIMARY** **KEY** **CLUSTERED**,

ClassID **int**,

AcademicHoursCount **int**

**CONSTRAINT** CK\_LessonPlan\_AcademicHoursCount **CHECK** (AcademicHoursCount > 0)

)

Создание внешних ключей:

**ALTER** **TABLE** Class

**ADD** **CONSTRAINT** FK\_Class\_Teacher

**FOREIGN** **KEY** (TeacherID)

**REFERENCES** Teacher (TeacherID);

**ALTER** **TABLE** Class

**ADD** **CONSTRAINT** FK\_Class\_Subject

**FOREIGN** **KEY** (SubjectID)

**REFERENCES** Subject (SubjectID)

**ALTER** **TABLE** Class

**ADD** **CONSTRAINT** FK\_Class\_Group

**FOREIGN** **KEY** (GroupID)

**REFERENCES** [Group] (GroupID)

**ALTER** **TABLE** ClassSchedule

**ADD** **CONSTRAINT** FK\_ClassSchedule\_Class

**FOREIGN** **KEY** (ClassID)

**REFERENCES** Class (ClassID);

**ALTER** **TABLE** ClassPlan

**ADD** **CONSTRAINT** FK\_ClassPlan\_Class

**FOREIGN** **KEY** (ClassID)

**REFERENCES** Class (ClassID);

Ниже приведен код, с помощью которого можно проверить работу ограничений, заданных нами:

**USE** Schedule

**INSERT** **INTO** [Group] (Course, GroupNumber, AmountOfStudents)

**VALUES** (1, 'Группа 58', 5)

**INSERT** **INTO** [Group] (Course, GroupNumber, AmountOfStudents)

**VALUES**(1, 'Группа 58', 35)

**INSERT** **INTO** [Group] (Course, GroupNumber, AmountOfStudents)

**VALUES**(1, 'Группа 1', 25)

**INSERT** **INTO** [Group] (Course, GroupNumber, AmountOfStudents)

**VALUES**(6, 'Группа 58', 20)

**INSERT** **INTO** Subject (SubjectName)

**VALUES**('Предмет 1')

**INSERT** **INTO** ClassPlan (ClassID, AcademicHoursCount)

**VALUES**(1, -5)

**INSERT** **INTO** ClassSchedule (ClassID, StartTime, EndTime, Auditorium)

**VALUES**(1, '08:15:00', '09:35:00')