Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ЗНАКОМСТВО С РЕЛЯЦИОННЫМ ПРОЕКТИРОВАНИЕМ ДАННЫХ

Отчет по индивидуальному заданию по дисциплине «Основы разработки баз данных»

Студент гр. 573-3

\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Макаров

дата

\_\_\_\_\_\_\_\_\_\_\_\_

подпись

Руководитель:

Преподаватель:

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ Р. О. Остапенко

подпись

оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

Томск, 2025

**Введение**

В современном образовании автоматизация управления учебным процессом играет ключевую роль в повышении эффективности и удобства работы преподавателей и администрации. Разработка базы данных для хранения информации об учениках, учителях, классах, расписании и оценках позволяет оптимизировать учет данных, минимизировать ошибки и упростить процесс формирования отчетности.

Цель данной работы — разработка концептуальной и реляционной модели базы данных для автоматизированной системы управления учебным процессом.

Результатом работы станет структурированная реляционная модель данных, обеспечивающая целостность, непротиворечивость и удобство работы с учебной информацией.

**Задачи:**

1. Определить отношения (таблицы) и связи между отношениями в предметной области на основе концептуальной информационной модели из индивидуального задания №1.

2. Определить атрибуты, а так же первичные и внешние ключи в отношениях (таблицах).

3. Подвергнуть полученные отношения процессу нормализации (использовать только первые три нормальные формы).

4. Полученный после нормализации результат представить в виде реляционной модели данных в методологии IDEF1x.

1. **ОСНОВНАЯ ЧАСТЬ**

**1.1 Определение отношений в таблицах для построения модели**

Для построения модели данных (рисунок 1.1), на основе концептуальной модели из прошлого задания, мы можем определить следующие сущности и связи между ними. Также, пересмотрев концептуальную модель из прошлого задания, я пришёл к решению, что хранить Отчёт отдельным атрибутом не имеет смысла, так как все поля являются вычисляемыми.

1. Учитель:

* ФИО
* Номер кабинета
* Список классов
* Предметы

Связи:

* Один учитель может преподавать многие предметы (1:M → Предмет)
* Один учитель может быть классным руководителем только одного класса (1:1 → Класс)

1. Ученик:

* ФИО
* Класс
* Список оценок

Связи:

* Один ученик принадлежит к одному классу (M:1 → Класс)
* Один ученик может получать много оценок (1:M → Оценка)

1. Класс:

* Номер и буква
* Классный руководитель
* Количество учеников
* Список учеников

Связи:

* Один класс может иметь много учеников (1:M → Ученик)
* Один класс имеет одно расписание (1:1 → Расписание)

1. Предмет:

* Название
* Учитель

Связи:

* Один предмет преподается многими учителями (M:M → Учитель)
* Один предмет может входить в много расписаний (1:M → Расписание)

1. Оценка:

* Ученик
* Оценка
* Четверть
* Предмет

Связи:

* Оценка относится к одному ученику (M:1 → Ученик)
* Оценка ставится по одному предмету (M:1 → Предмет)

1. Расписание:

* Класс
* День недели
* Номер урока
* Кабинет
* Предмет

Связи:

* Расписание принадлежит одному классу (1:1 → Класс)
* Расписание содержит много предметов (1:M → Предмет)

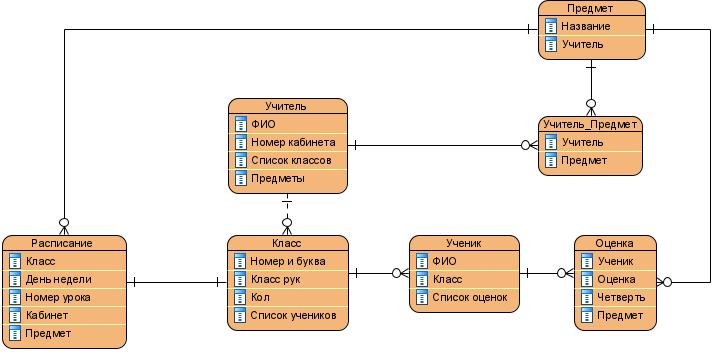


Рисунок 1.1 – Исходная модель данных

* 1. **Выделение первичных и внешних значений**

Для представления данных в реляционной модели, необходимо выразить первичные ключи и внешние ключи:

1. Учитель (Teacher)

* PK: teacher\_id
* ФИО
* Номер кабинета
* Преподаваемые предметы

2. Класс (Class)

* PK: class\_id
* Номер и буква
* Список учеников
* FK: teacher\_id (классный руководитель)

3. Ученик (Student)

* PK: student\_id
* ФИО
* Список оценок
* FK: class\_id (принадлежит классу)

4. Предмет (Subject)

* PK: subject\_id
* Название
* FK: teacher\_id (Учитель)

5. Оценка (Grade)

* PK: grade\_id
* FK: student\_id (Ученик)
* FK: Название предмета (предмет)
* Четверть
* Оценка

6. Расписание (Schedule)

* PK: schedule\_id
* FK: class\_id (класс)
* FK: subject\_id (предмет)
* День недели
* Номер урока
* Номер кабинета

После выделения первичных и внешних ключей мы можем представить нашу схему в нотации IDEF1x (рисунок 1.2)

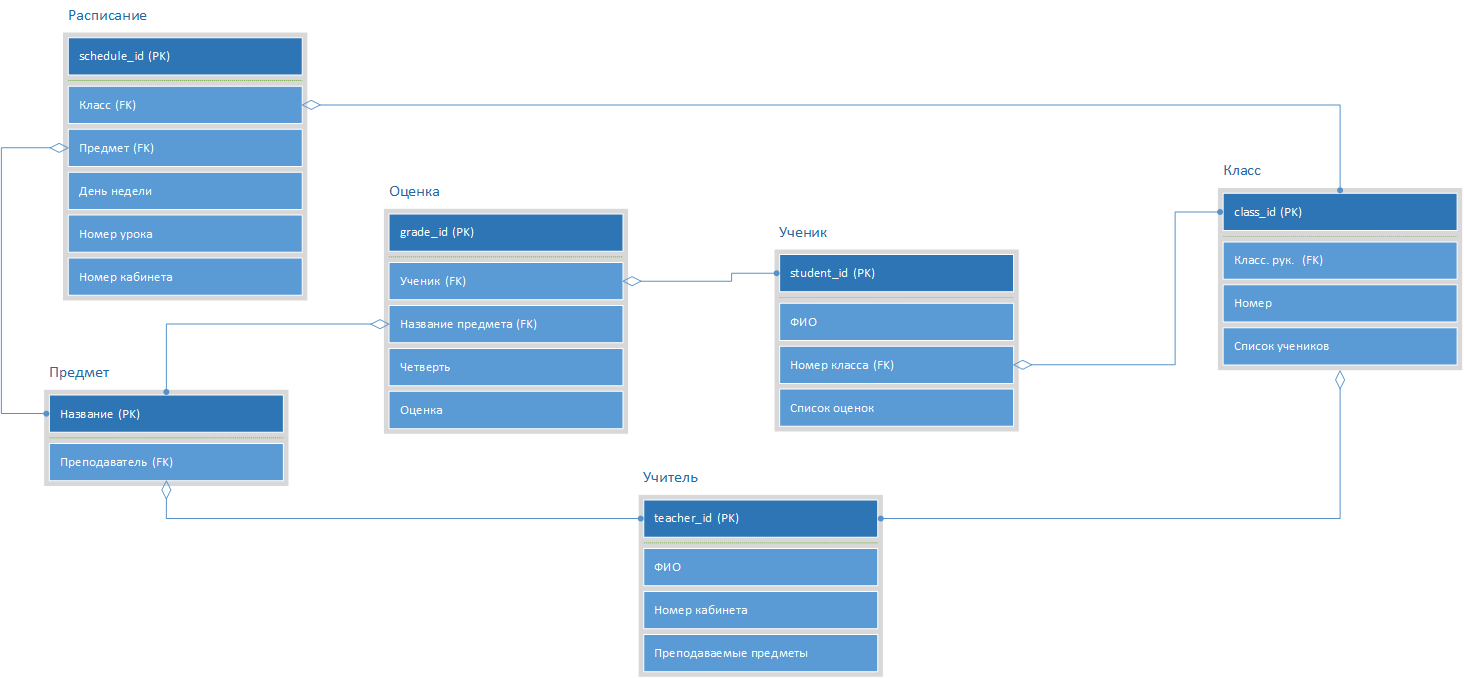


Рисунок 1.2 – Исходная модель данных в нотации IDEF1x

* 1. **Нормализация**

Нормализация – процесс реорганизации данных путем ликвидации повторяющихся групп и иных противоречий в хранении данных с целью приведения таблиц к виду, позволяющему осуществлять непротиворечивое и корректное редактирование данных.

Чтобы нормализовать предложенные данные по первой нормальной форме (1NF), нужно выполнить несколько шагов. Первая нормальная форма требует, чтобы все атрибуты таблицы содержали только атомарные значения, то есть каждый столбец должен содержать одно значение (не массив или список).

Изменения:

* Переносим связь "учитель-преподаваемые предметы" в отдельную таблицу, чтобы каждый учитель мог преподавать несколько предметов.

Преподаваемые предметы (Teacher\_Subject):

* teacher\_id (FK)
* subject\_id (FK) (ссылается на Предмет)
* Разобьём номер и букву класса для атомарности.
* Создаем отдельную таблицу для оценок ученика.

Оценки ученика (Student\_Grade):

* student\_id (FK)
* Название (FK) (ссылается на Предмет)
* Четверть
* Оценка

Чтобы привести данные ко второй нормальной форме (2NF), необходимо выполнить два основных шага:

* Удалить частичные зависимости, т.е. все атрибуты должны зависеть от всего первичного ключа, а не только от его части.
* Обеспечить, чтобы каждая таблица имела только одну зависимость от ключа. Если атрибут зависит от части составного ключа (если он существует), его нужно перенести в другую таблицу.

Таблица Оценки ученика (Student\_Grade) содержит частичные зависимости. Атрибут Оценка зависит от сочетания student\_id и subject\_name, а Четверть тоже зависит от student\_id и subject\_name. Однако, Четверть и Оценка не зависят от полного составного ключа, если мы рассматриваем составной ключ как student\_id и subject\_name.

Чтобы устранить частичные зависимости, нужно переместить информацию о четверти в отдельную таблицу:

Оценки ученика (Student\_Grade):

* PK: grade\_id
* FK: student\_id (ссылается на Ученик)
* FK: Название (ссылается на Предмет)
* FK: Номер четверти (ссылается на Четверть)
* Оценка

Четверти (Quarter):

* PK: quarter\_id
* Номер четверти

Расписание (Schedule)

* PK: schedule\_id
* FK: class\_id (класс)
* FK: quarter\_id (четверть)
* FK: subject\_id (предмет)
* День недели
* Номер урока
* Номер кабинета

Чтобы привести данные к третьей нормальной форме (3NF), нужно выполнить два шага:

* Удалить транзитивные зависимости, т.е. атрибуты, которые зависят от других атрибутов, а не от первичного ключа.
* Обеспечить, чтобы все атрибуты в таблице зависели напрямую от первичного ключа и не имели зависимости от других неключевых атрибутов.

В нашем случае, мы будем искать атрибуты, которые зависят от других атрибутов, и переработаем структуру, чтобы устранить транзитивные зависимости.

В процессе преобразования во вторую нормальную форму, мы автоматически перешли в третью, поскольку все транзитивные зависимости перешли в другие таблицы.

По итогам трёх форм нормализации мы можем построить реляционную модель данных в методологии IDEF1x (рисунок 1.3).

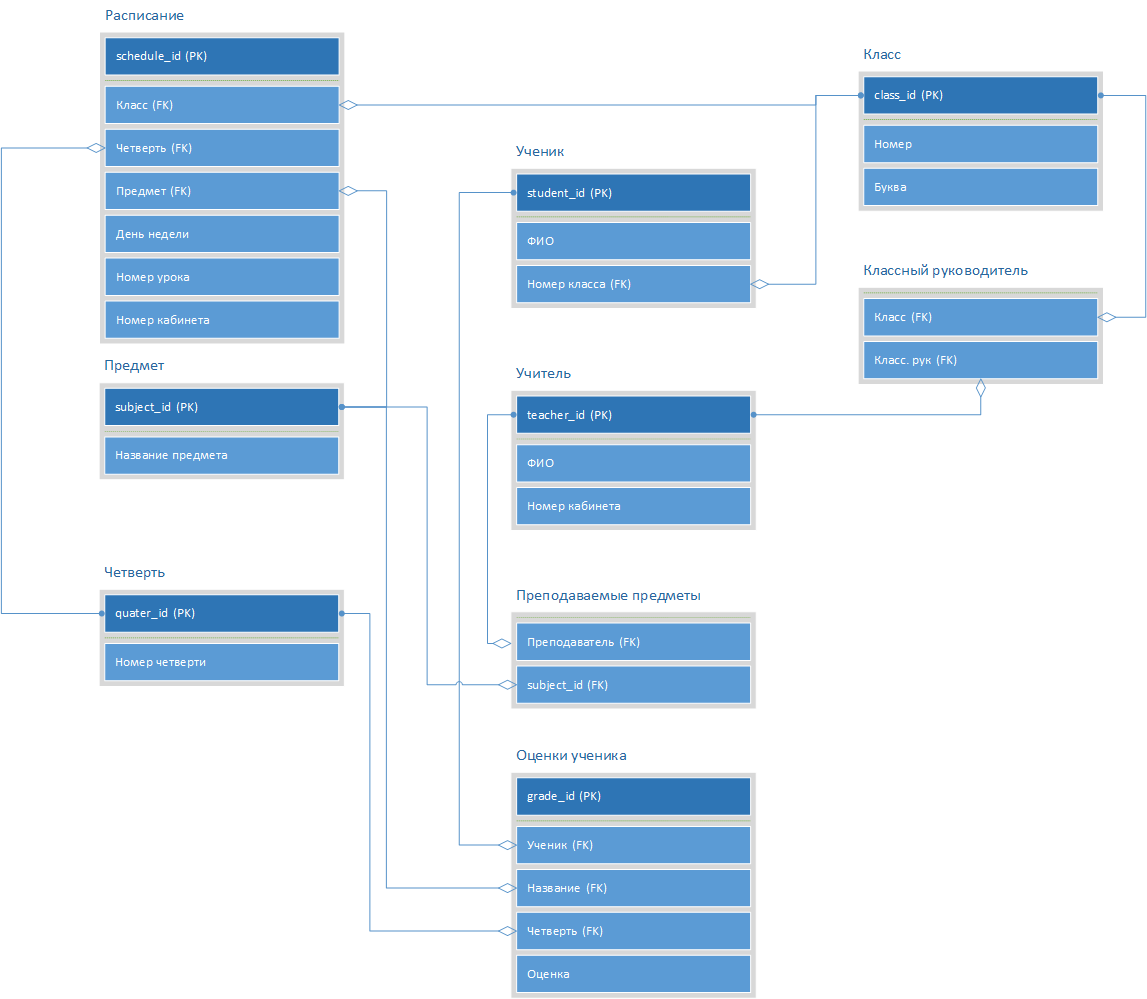


Рисунок 1.3 – Реляционная модель данных в методологии IDEF1x

**ЗАКЛЮЧЕНИЕ**

В ходе работы была разработана модель базы данных для автоматизированной системы управления учебным процессом. Определены основные сущности, их атрибуты и связи. Затем были выделены отношения (таблицы) с указанием первичных и внешних ключей, что обеспечило логическую связанность данных.

Для улучшения структуры базы данных был применен процесс нормализации до третьей нормальной формы (3NF). Это позволило устранить избыточность данных, минимизировать аномалии при обновлении и повысить целостность информации.

Разработанная база данных обеспечивает удобное хранение и обработку информации об учениках, учителях, классах, оценках и расписании.