

Доклад

Сетевые модели

Коннова Т. А.

2025-05-03

Российский университет дружбы народов, Москва, Россия

Информация

- Коннова Татьяна Алексеевна
- студент
- Российский университет дружбы народов



Вводная часть

- Целью работы по теме сетевых моделей является исследование различных подходов и методов построения сетей для оптимизации передачи данных и повышения эффективности управления сетевыми ресурсами.

- В современном мире сетевые технологии играют ключевую роль в обеспечении связи, обработки данных и обмена информацией. Сетевые модели служат основой для проектирования и реализации компьютерных сетей, позволяя структурировать сложные системы и оптимизировать процессы передачи данных.

Принципы работы сетевых моделей с примерами Сетевые модели, такие как OSI и TCP/IP, организуют взаимодействие между компонентами сетей, используя несколько ключевых принципов:

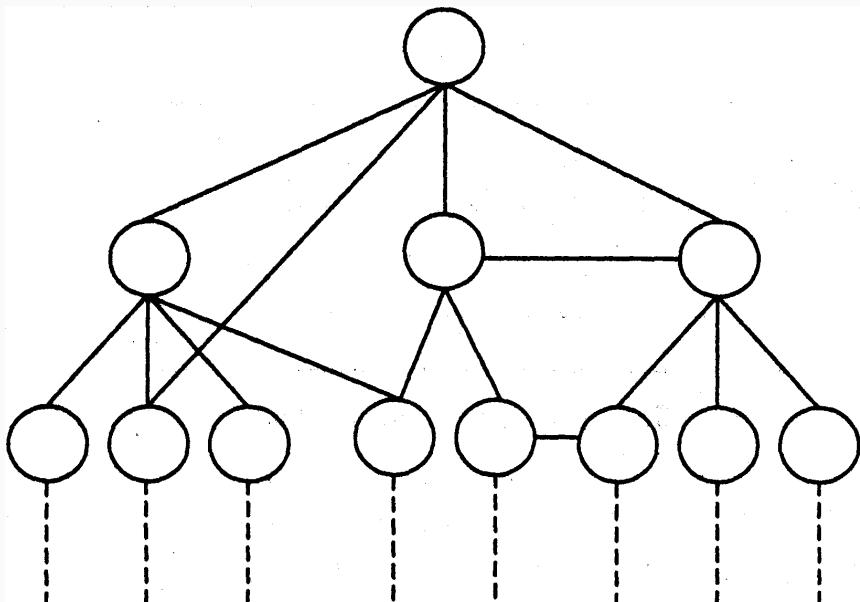
- Модульность
- Интерфейсы
- Стандартизация
- Абстракция
- Службы и функции

Разница между иерархической моделью данных и сетевой моделью заключается в структуре и способах организации данных. В иерархических структурах каждая запись-потомок должна иметь ровно одного предка, что создает строгую иерархию, подобную древовидной структуре. Это означает, что данные организованы в виде единого дерева, где каждое узловое значение имеет только одну “родительскую” запись, что ограничивает гибкость в представлении сложных взаимосвязей.

Сетевая модель была одним из первых подходов, использовавшимся при создании баз данных в конце 50-х — начале 60-х годов. Активным пропагандистом этой модели был Чарльз Бахман. Идеи Бахмана послужили основой для разработки стандартной сетевой модели под эгидой организации CODASYL.



Сетевая СУБД — это СУБД, построенная на основе сетевой модели данных. К основным понятиям сетевой модели базы данных относятся: уровень, элемент (узел), связь. Узел — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. В сетевой структуре каждый элемент может быть связан с любым другим элементом.



1. Определение сущностей

- **Студенты (S):** Каждый студент имеет уникальный идентификатор, имя и другие атрибуты (например, дата рождения, специальность).
- **Курсы (C):** Каждый курс также имеет уникальный идентификатор, название, описание и количество кредитов.
- **Преподаватели (T):** Каждый преподаватель имеет уникальный идентификатор, имя и предметы, которые он преподает.

2. Определение связей

- **Запись на курс (R):** Связь между студентами и курсами, показывающая, какие студенты записаны на какие курсы. Один студент может быть записан на несколько курсов, а один курс может иметь множество студентов.
- **Преподавание (P):** Связь между преподавателями и курсами, показывающая, какие преподаватели ведут какие курсы. Один преподаватель может вести несколько курсов, а один курс может быть веден несколькими преподавателями.

3. Структура данных

- **Студенты (S):**
 - ID_студента (уникальный)
 - Имя
 - Специальность
- **Курсы (C):**
 - ID_курса (уникальный)
 - Название
 - Описание
- **Преподаватели (T):**
 - ID_преподавателя (уникальный)
 - Имя
 - Предметы
- **Связи:**
 - **Запись на курс (R):**
 - ID_студента
 - ID_курса
 - **Преподавание (P):**
 - ID_преподавателя
 - ID_курса

4. Пример данных

- **Студенты:**

- (1, "Иванов Иван", "Информатика")
- (2, "Петрова Анна", "Математика")
- (3, "Сидоров Алексей", "Физика")

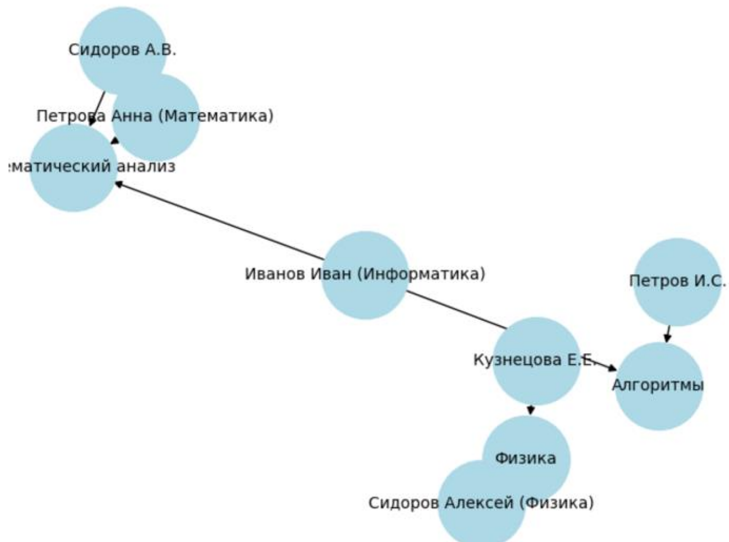
- **Курсы:**

- (101, "Алгоритмы", "Изучение алгоритмов и структур данных")
- (102, "Математический анализ", "Основы математического анализа")
- (103, "Физика", "Основы физики")

Рис. 5: img.png

- **Преподаватели:**
 - (201, "Петров И.С.", "Алгоритмы")
 - (202, "Сидоров А.В.", "Математический анализ")
 - (203, "Кузнецова Е.Е.", "Физика")
- **Связи:**
 - **Запись на курс (R):**
 - (1, 101)
 - (1, 102)
 - (2, 102)
 - (3, 103)
 - **Преподавание (P):**
 - (201, 101)
 - (202, 102)
 - (203, 103)

Модель “Студенты, Курсы и Преподаватели”



- Сетевые модели данных представляют собой важный этап в эволюции систем управления базами данных. Они позволяют эффективно организовывать и представлять данные в виде узлов и связей, что обеспечивает гибкость и возможность сложных запросов. Одним из главных преимуществ сетевых моделей является их способность обрабатывать сложные отношения между данными, что делает их подходящими для приложений с высокой степенью взаимосвязанности.