

Уральский государственный экономический университет
ФОРМАЛИЗАЦИЯ ИНФОРМАЦИИ И Big Data

02.03.03 -Математическое обеспечение и администрирование информационных систем,
направленность (профиль) -разработка и администрирование информационных систем

<http://vikchas.ru>

**Лабораторная работа 1. Тема «основные
модели представления и
структурирования данных»**

Часовских Виктор Петрович
доктор технических наук, профессор кафедры
ШИиКМ, ФГБОУ ВО «Уральский государственный
экономический университет

Екатеринбург 2026

Цель работы. Изучить основные модели представления и структурирования данных, научиться выбирать подходящую модель данных для конкретной предметной области и формализовать данные в табличной, иерархической, графовой и JSON-структуре.

В ходе выполнения работы необходимо:

1. Изучить основные модели представления данных.
2. Рассмотреть особенности структурированных, полуструктурированных и неструктурированных данных.
3. Построить табличную модель данных.
4. Построить иерархическую модель данных.
5. Построить графовую модель данных.
6. Представить данные в формате JSON.
7. Сравнить модели представления данных и определить их преимущества и ограничения.

Данные — это зафиксированные сведения об объектах, событиях, процессах или явлениях, пригодные для хранения, передачи и обработки.

Модель данных — это способ описания структуры данных, связей между ними и правил их обработки.

JSON (JavaScript Object Notation) — это текстовый формат обмена структурированными данными. Он широко применяется в веб-разработке, API, конфигурационных файлах и

Пример «Представление данных в формате JSON»

В данном примере информация об учебном процессе представлена в формате **JSON** (JavaScript Object Notation.)

JSON — это текстовый формат обмена данными, который позволяет хранить информацию в виде пар «**ключ — значение**», а также использовать массивы и вложенные объекты.

Для описания предметной области были использованы следующие объекты:

- учебное заведение;
- группы;
- студенты;
- дисциплины;
- преподаватели;
- оценки.

Структура JSON построена иерархически:
сначала указывается учебное заведение, затем список групп, внутри каждой группы — список студентов, а у каждого студента — список изучаемых дисциплин с преподавателем и оценкой.

JSON-представление данных

JSON

```
{
  "college": "Институт информационных технологий",
  "groups": [
    {
      "group_name": "ИС-21",
      "students": [
        {
          "student_id": 1,
          "name": "Иванов Иван",
          "disciplines": [
            {
              "discipline_name": "Базы данных",
              "teacher": "Смирнова А.А.",
              "grade": 5
            },
            {
              "discipline_name": "Программирование",
              "teacher": "Орлова Н.Н.",
              "grade": 4
            }
          ]
        },
        {
          "student_id": 2,
          "name": "Петров Пётр",
          "disciplines": [
            {
              "discipline_name": "Базы данных",
              "teacher": "Смирнова А.А.",
              "grade": 4
            },
            {
              "discipline_name": "Big Data",
              "teacher": "Кузнецов В.В.",
              "grade": 5
            }
          ]
        }
      ]
    },
    {
      "group_name": "ИС-22",
      "students": [
        {
```

```

    "student_id": 3,
    "name": "Сидорова Анна",
    "disciplines": [
      {
        "discipline_name": "Big Data",
        "teacher": "Кузнецов В.В.",
        "grade": 5
      },
      {
        "discipline_name": "Программирование",
        "teacher": "Орлова Н.Н.",
        "grade": 5
      }
    ]
  },
  {
    "student_id": 4,
    "name": "Кузнецов Алексей",
    "disciplines": [
      {
        "discipline_name": "Базы данных",
        "teacher": "Смирнова А.А.",
        "grade": 3
      },
      {
        "discipline_name": "Программирование",
        "teacher": "Орлова Н.Н.",
        "grade": 4
      }
    ]
  }
]
}
]
}
]
}
}

```

Пояснение к структуре JSON

Элемент JSON	Описание
"college"	Название учебного заведения
"groups"	Массив учебных групп
"group_name"	Название группы
"students"	Массив студентов внутри группы
"student_id"	Уникальный идентификатор студента
"name"	ФИО студента
"disciplines"	Массив изучаемых дисциплин
"discipline_name"	Название дисциплины

Элемент JSON	Описание
"teacher"	Преподаватель дисциплины
"grade"	Оценка студента по дисциплине

Данные об учебном процессе были представлены в формате JSON. Такой формат удобен для хранения вложенной информации, так как позволяет описывать группы, студентов, дисциплины и оценки в одной структурированной записи. JSON легко читается человеком и широко используется в веб-приложениях, API и документно-ориентированных базах данных.

Табличная модель

Табличная модель представляет данные в виде таблиц, состоящих из строк и столбцов.

Пример:

ID	ФИО студента	Группа	Дисциплина	Оценка
1	Иванов И.И.	ИС-21	Базы данных	5
2	Петров П.П.	ИС-21	Big Data	4

Преимущества:

- простота восприятия;
- удобство хранения структурированных данных;
- возможность использования SQL.

Недостатки:

- сложно описывать вложенные структуры;
- ограниченная гибкость при изменении структуры данных.

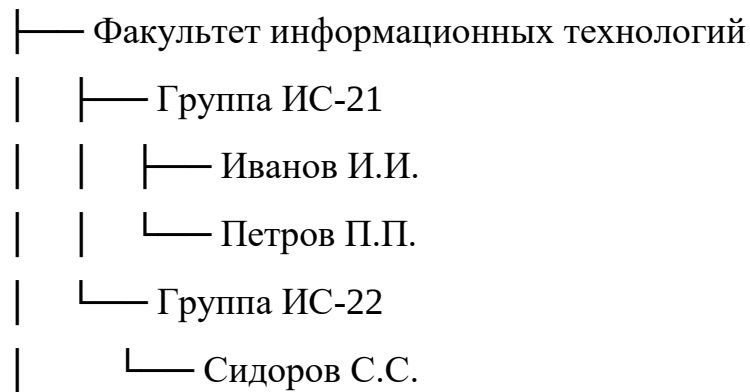
Иерархическая модель

Иерархическая модель представляет данные в виде дерева, где каждый элемент имеет родителя и дочерние элементы.

Пример:

ТЕХТ

Университет



Преимущества:

- наглядное отображение подчинённости;
- удобна для организационных структур, каталогов, файловых систем.

Недостатки:

- сложно описывать связи «многие ко многим»;
- изменение структуры может быть трудоёмким.

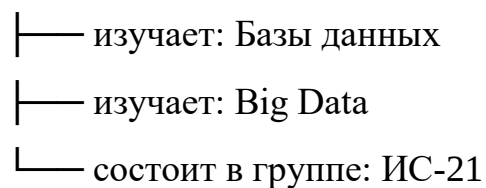
Сетевая модель

Сетевая модель позволяет одному объекту иметь несколько связей с другими объектами.

Пример:

ТЕХТ

Студент Иванов И.И.



Преимущества:

- позволяет описывать сложные связи;
- подходит для связанных данных.

Недостатки:

- сложнее для понимания и реализации;
- требует явного описания связей.

Графовая модель

Графовая модель представляет данные в виде вершин и рёбер.

- **Вершины** — объекты.
- **Рёбра** — связи между объектами.

Пример:

ТЕХТ

[Студент: Иванов] --изучает--> [Дисциплина: Big Data]

[Студент: Иванов] --состоит в--> [Группа: ИС-21]

[Преподаватель: Смирнова] --ведёт--> [Дисциплина: Big Data]

Преимущества:

- удобна для анализа связей;
- используется в социальных сетях, рекомендательных системах, Big Data.

Недостатки:

- требует специальных инструментов для эффективной обработки;
- сложнее традиционных таблиц.

Документно-ориентированная модель

Данные хранятся в виде документов, например JSON или XML.

Пример JSON:

JSON

```
{
  "student_id": 1,
  "name": "Иванов Иван Иванович",
  "group": "ИС-21",
  "disciplines": [
    {
      "name": "Базы данных",
      "grade": 5
    },
    {
      "name": "Big Data",
      "grade": 4
    }
  ]
}
```

}

Преимущества:

- гибкая структура;
- поддержка вложенных данных;
- удобна для веб-приложений и NoSQL БД.

Недостатки:

- возможна избыточность данных;
- сложнее обеспечивать целостность данных.

Исходные данные

Используется предметная область: «**Учебный процесс в Институте**».

Необходимо описать следующие объекты:

- студент;
- группа;
- дисциплина;
- преподаватель;
- оценка.

Минимальный набор данных:

Студент	Группа	Дисциплина	Преподаватель	Оценка
Иванов Иван	ИС-21	Базы данных	Смирнова А.А.	5
Иванов Иван	ИС-21	Big Data	Кузнецов В.В.	4
Петров Пётр	ИС-21	Базы данных	Смирнова А.А.	4
Сидорова Анна	ИС-22	Big Data	Кузнецов В.В.	5
Сидорова Анна	ИС-22	Программирование	Орлова Н.Н.	5

Порядок выполнения работы

Задание 1. Определение объектов предметной области

1. Ознакомьтесь с исходными данными.
2. Определите основные сущности предметной области.
3. Для каждой сущности выделите атрибуты.

Пример:

Сущность	Атрибуты
Студент	ID студента, ФИО, группа
Группа	ID группы, название группы
Дисциплина	ID дисциплины, название
Преподаватель	ID преподавателя, ФИО
Оценка	ID студента, ID дисциплины, оценка

Задание 2. Построение табличной модели данных

Создайте табличную модель данных.

Вариант 1. Одна общая таблица

ID	Студент	Группа	Дисциплина	Преподаватель	Оценка
1	Иванов Иван	ИС-21	Базы данных	Смирнова А.А.	5
2	Иванов Иван	ИС-21	Big Data	Кузнецов В.В.	4

Вариант 2. Нормализованная структура

Создайте несколько таблиц:

Таблица «Студенты»

student_id	ФИО	group_id
1	Иванов Иван	1
2	Петров Пётр	1
3	Сидорова Анна	2

Таблица «Группы»

group_id	Группа
1	ИС-21
2	ИС-22

Таблица «Дисциплины»

discipline_id	Дисциплина
1	Базы данных
2	Big Data
3	Программирование

Таблица «Преподаватели»

teacher_id	Преподаватель
1	Смирнова А.А.
2	Кузнецов В.В.
3	Орлова Н.Н.

Таблица «Оценки»

student_id	discipline_id	teacher_id	Оценка
1	1	1	5

student_id	discipline_id	teacher_id	Оценка
1	2	2	4
2	1	1	4
3	2	2	5
3	3	3	5

Задание 3. Построение иерархической модели

Представьте исходные данные в виде дерева.

Пример:

ТЕХТ

Учебное заведение



Необходимо:

1. Построить иерархию.
2. Определить корневой элемент.
3. Определить уровни иерархии.

Задание 4. Построение графовой модели

Постройте графовую модель данных.

Вершины графа:

- студенты;
- группы;
- дисциплины;
- преподаватели.

Рёбра графа:

- студент **состоит в** группе;
- студент **изучает** дисциплину;
- преподаватель **ведёт** дисциплину;
- студент **получил оценку** по дисциплине.

Пример описания:

ТЕХТ

[Иванов Иван] --состоит в--> [ИС-21]

[Иванов Иван] --изучает--> [Базы данных]

[Иванов Иван] --получил оценку 5--> [Базы данных]

[Смирнова А.А.] --ведёт--> [Базы данных]

Результат можно представить:

- в текстовом виде;
- в виде схемы;
- в draw.io / diagrams.net;
- в виде таблицы связей.

Пример таблицы связей:

Объект 1	Связь	Объект 2
Иванов Иван	состоит в	ИС-21
Иванов Иван	изучает	Базы данных

Объект 1	Связь	Объект 2
Иванов Иван	получил оценку 5	Базы данных
Смирнова А.А.	ведёт	Базы данных

Задание 5. Представление данных в формате JSON

Опишите данные о студентах в формате JSON.

Пример:

JSON

```
{
  "college": "Учебное заведение",
  "groups": [
    {
      "group_name": "ИС-21",
      "students": [
        {
          "student_id": 1,
          "name": "Иванов Иван",
          "disciplines": [
            {
              "name": "Базы данных",
              "teacher": "Смирнова А.А.",
              "grade": 5
            },
            {
              "name": "Big Data",
              "teacher": "Кузнецов В.В.",
              "grade": 4
            }
          ]
        },
        {
          "student_id": 2,
          "name": "Петров Пётр",
          "disciplines": [
            {
              "name": "Базы данных",
              "teacher": "Смирнова А.А.",
              "grade": 4
            }
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
},
{
  "group_name": "ИС-22",
  "students": [
    {
      "student_id": 3,
      "name": "Сидорова Анна",
      "disciplines": [
        {
          "name": "Big Data",
          "teacher": "Кузнецов В.В.",
          "grade": 5
        },
        {
          "name": "Программирование",
          "teacher": "Орлова Н.Н.",
          "grade": 5
        }
      ]
    }
  ]
}
]
}
]
}
]
}

```

Проверьте корректность JSON с помощью онлайн-валидатора.

Задание 6. Сравнение моделей данных

Заполните сравнительную таблицу.

Модель данных	Достоинства	Недостатки	Где применяется
Табличная	Простота, SQL, строгая структура	Сложно хранить вложенные данные	Реляционные БД
Иерархическая	Наглядность, подчинённость	Сложно описывать связи многие-ко-многим	Каталоги, XML

Модель данных	Достоинства	Недостатки	Где применяется
Графовая	Хорошо описывает связи	Требует специальных инструментов	Соцсети, рекомендации
Документная JSON	Гибкость, вложенность	Возможна избыточность	NoSQL, веб-сервисы

Требования к отчёту

Отчёт по лабораторной работе должен содержать:

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Выполненные задания:
 - список сущностей и атрибутов;
 - табличная модель;
 - иерархическая модель;
 - графовая модель;
 - JSON-представление;
 - сравнительная таблица моделей.
5. Ответы на контрольные вопросы.
6. Вывод по работе.

Контрольные вопросы

1. Что такое модель данных?
2. Для чего используется формализация информации?
3. Чем данные отличаются от информации?
4. Какие существуют основные модели представления данных?
5. В чём особенность табличной модели данных?

6. Что такое сущность и атрибут?
7. Что такое первичный ключ?
8. Что такое внешний ключ?
9. Чем нормализованная таблица отличается от одной общей таблицы?
10. Какие преимущества имеет иерархическая модель данных?
11. Какой главный недостаток иерархической модели?
12. Что такое графовая модель данных?
13. Что является вершинами и рёбрами графа?
14. Где применяются графовые модели данных?
15. Что такое JSON?
16. Почему JSON относят к полуструктурированным данным?
17. Какие преимущества имеет документно-ориентированная модель?
18. Какие недостатки есть у документной модели?
19. Какую модель удобнее использовать для хранения информации о студентах и оценках?
20. Какую модель данных лучше использовать для анализа сложных связей между объектами?

Пример вывода

В ходе лабораторной работы были изучены основные модели представления и структурирования данных. Были построены табличная, иерархическая, графовая и документная модели для предметной области «Учебный процесс». В результате работы были закреплены навыки формализации информации, выделения сущностей и атрибутов, описания связей между объектами и представления данных в формате JSON.