

Задание 1. Описание структур данных.

1. Иерархическая модель данных

Предметная область: Система управления файлами в операционной системе.

Описание:

- Объекты: Каталоги и файлы.
- Взаимоотношения: Один каталог может содержать подкаталоги и файлы (один-ко-многим). Каждый файл принадлежит только одному каталогу.
- Структура: Дерево (иерархия) — родитель → потомки.

Пример:

Корень

| - Папка_Документы

| | - Файл1.docx

| | - Файл2.pdf

| - Папка_Изображения

 | - Фото1.jpg

2. Сетевая модель данных

Предметная область: Управление поставками в производственной компании.

Описание:

- Объекты: Поставщики, компоненты, заказы.
- Взаимоотношения: Компонент может поставляться несколькими поставщиками, и один поставщик может поставлять множество компонентов.
- Структура: Граф с произвольными связями между узлами.

Пример:

- Поставщик А поставляет Компонент 1 и Компонент 2.
- Компонент 2 поставляется также Поставщиком В.

3. Реляционная модель данных

Предметная область: Интернет-магазин.

Описание:

- Объекты: Клиенты, заказы, товары.
- Взаимоотношения:
 - Один клиент может сделать много заказов.
 - Один заказ может содержать множество товаров, и каждый товар может входить в разные заказы.

Таблицы:

- Клиенты(id, имя, email)
- Товары(id, название, цена)
- Заказы(id, клиент_id, дата)
- Состав_заказа(заказ_id, товар_id, количество)

4. Объектно-ориентированная модель

Предметная область: Система автоматизации библиотек.

Описание:

- Объекты: Книга, Автор, Читатель, Библиотекарь (представлены как классы с методами).
- Взаимоотношения: Наследование (например, Пользователь — базовый класс для Читатель и Библиотекарь), ассоциации (Книга связана с Автором, Читатель может брать Книгу).
- Модель включает не только данные, но и методы для работы с ними.

5. Документно-ориентированная модель (NoSQL)

Предметная область: Хранение профилей пользователей в социальной сети.

Описание:

- Объекты: Пользователи.
- Взаимоотношения: Каждый пользователь хранится как самостоятельный документ со всеми вложенными данными (имя, возраст,

список друзей, посты и т.д.).

- Структура: JSON-документы, вложенные структуры.

Пример документа:

```
{
  "id": 123,
  "имя": "Анна",
  "друзья": [456, 789],
  "посты": [
    {"текст": "Привет!", "дата": "2025-06-01"},
    {"текст": "Как дела?", "дата": "2025-06-03"}
  ]
}
```

6. Графовая модель данных (NoSQL)

Предметная область: Социальная сеть.

Описание:

- Объекты: Пользователи (вершины), Дружба или Подписка (ребра).
- Взаимоотношения: Много-много — каждый пользователь может быть связан с любым другим (напр., друзья, подписки, лайки).
- Используется для анализа взаимосвязей, рекомендаций, поиска путей.

Пример:

(Анна) —друг—> (Борис) —друг—> (Катя)

Задание 2. Таблица недостатков и преимуществ.

№	Модель данных	Преимущества	Недостатки
		— Простая структура	— Жесткая структура
1	Иерархическая	— Быстрый доступ при фиксированной иерархии	— Трудно реализовать связи многие-ко-многим
2	Сетевая	— Гибче иерархической	— Сложная навигация
		— Поддержка сложных связей (многие-ко-многим)	— Сложность модификации схемы

№	Модель данных	Преимущества	Недостатки
			данных
3	Реляционная	<ul style="list-style-type: none"> — Простота и гибкость— Стандарты SQL — Теоретическая основа (РКНФ) — Прямая связь с ООП-языками 	<ul style="list-style-type: none"> — Может быть медленнее при больших объемах — Ограничения при вложенности — Сложность в реализации
4	Объектно-ориентированная	<ul style="list-style-type: none"> — Наследование и инкапсуляция 	<ul style="list-style-type: none"> — Отсутствие универсального языка запросов
5	Документно-ориентированная (NoSQL)	<ul style="list-style-type: none"> — Гибкая структура — Высокая производительность при работе с JSON/XML — Идеальна для хранения связей 	<ul style="list-style-type: none"> — Отсутствие строгой схемы — Сложности при связях между документами — Нетрадиционная модель
6	Графовая (NoSQL)	<ul style="list-style-type: none"> — Эффективна для поиска путей и связей 	<ul style="list-style-type: none"> — Требуется специфических знаний и инструментов