

## **Реляционная Особенности**

- Организует данные в виде двухмерных таблиц.
- Используют структурированный язык запросов (SQL) для запросов.
- ACID - свойства

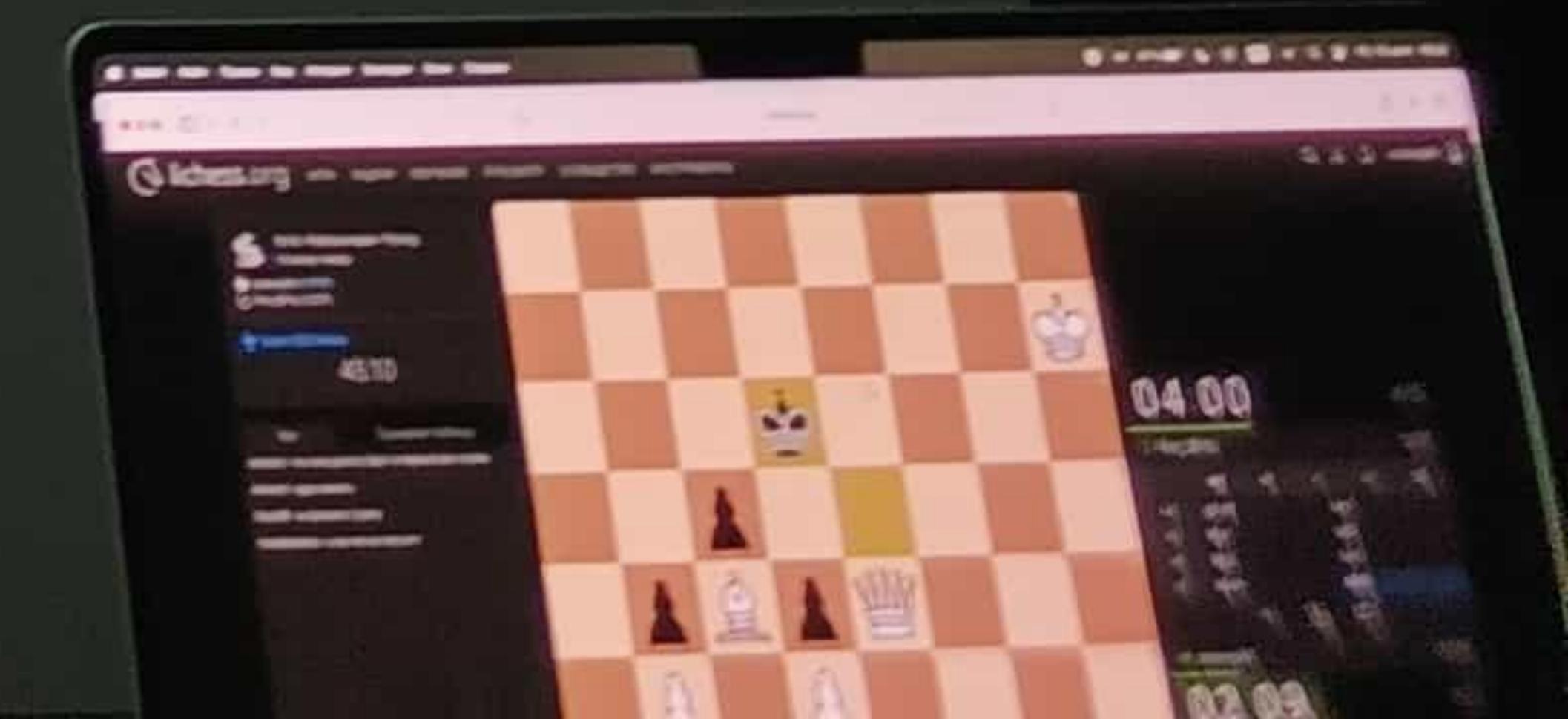
## Реляционная Преимущества

- Благодаря механизму ключей, ограничений и транзакций обеспечивается целостность и непротиворечивость данных.
- Использование SQL как стандарта облегчает разработку и поддержку приложений, а также миграцию данных между системами.
- Процесс нормализации позволяет устраниить избыточность данных и аномалии



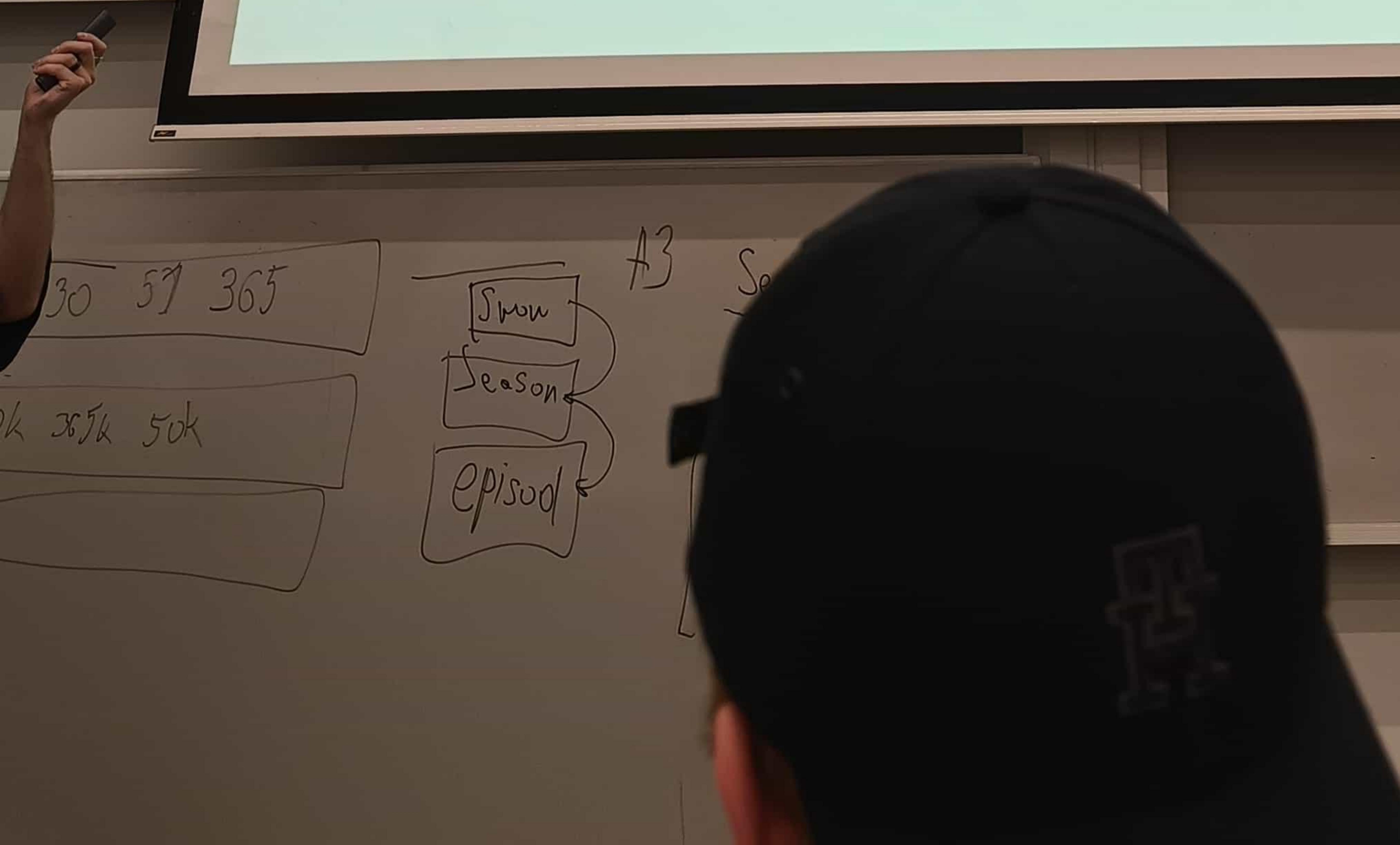
## Реляционная Недостатки

- Изменение структуры таблиц на работающей системе может быть сложной и дорогой операцией.
- Горизонтальное масштабирование реляционных баз данных зачастую сложнее реализовать по сравнению с некоторыми NoSQL решениями
- Сложные JOIN-запросы на огромных таблицах могут выполняться медленно.
- Неэффективна для хранения иерархических (древовидных) или графовых данных.



## Документоориентированная Преимущества

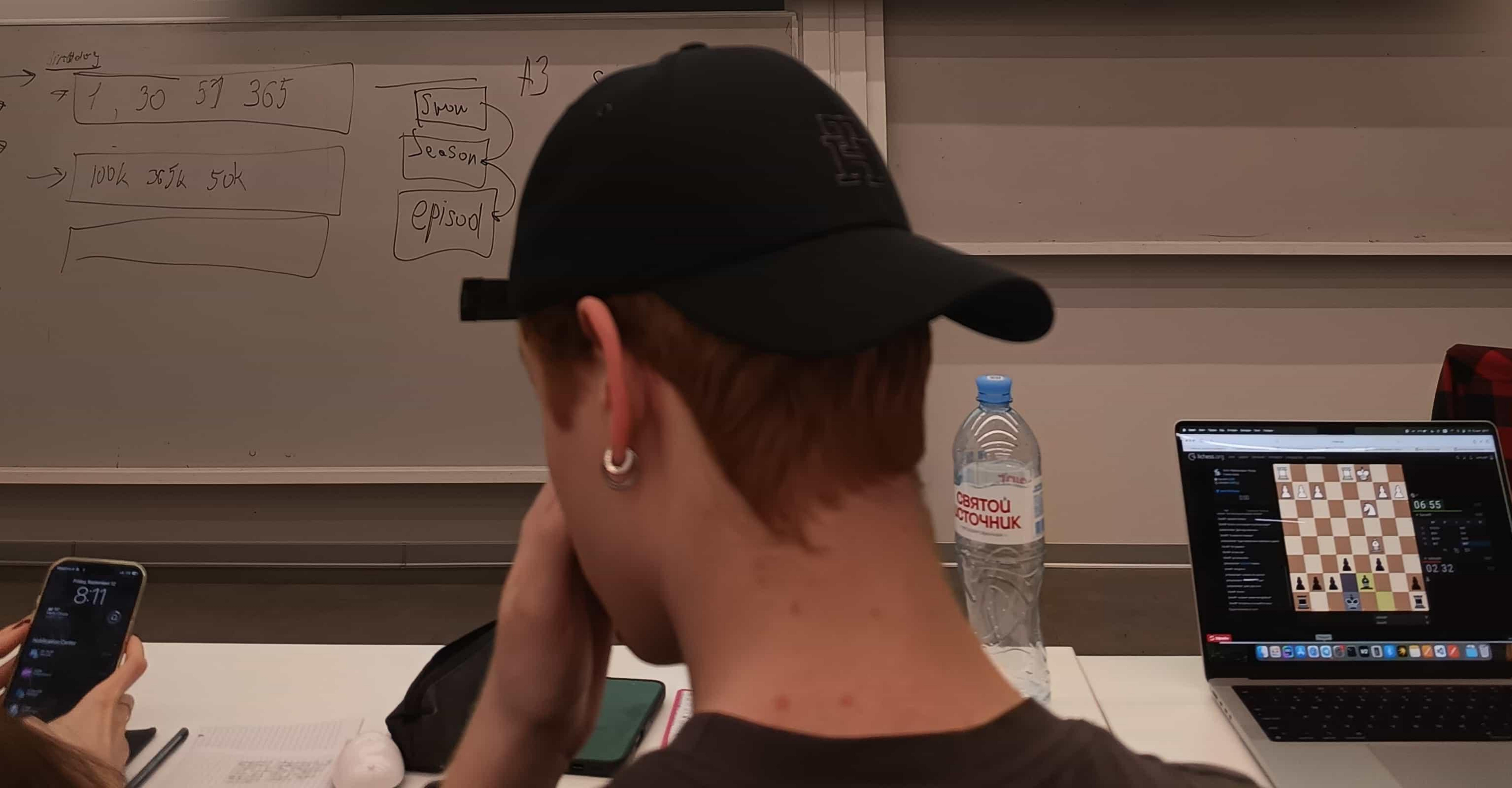
- Возможность изменения структуры документов без необходимости модифицировать общую схему базы данных
- Документы в формате JSON или BSON естественным образом отображаются на объекты в ЯП
- Простота горизонтальной масштабируемости



## Документоориентированная

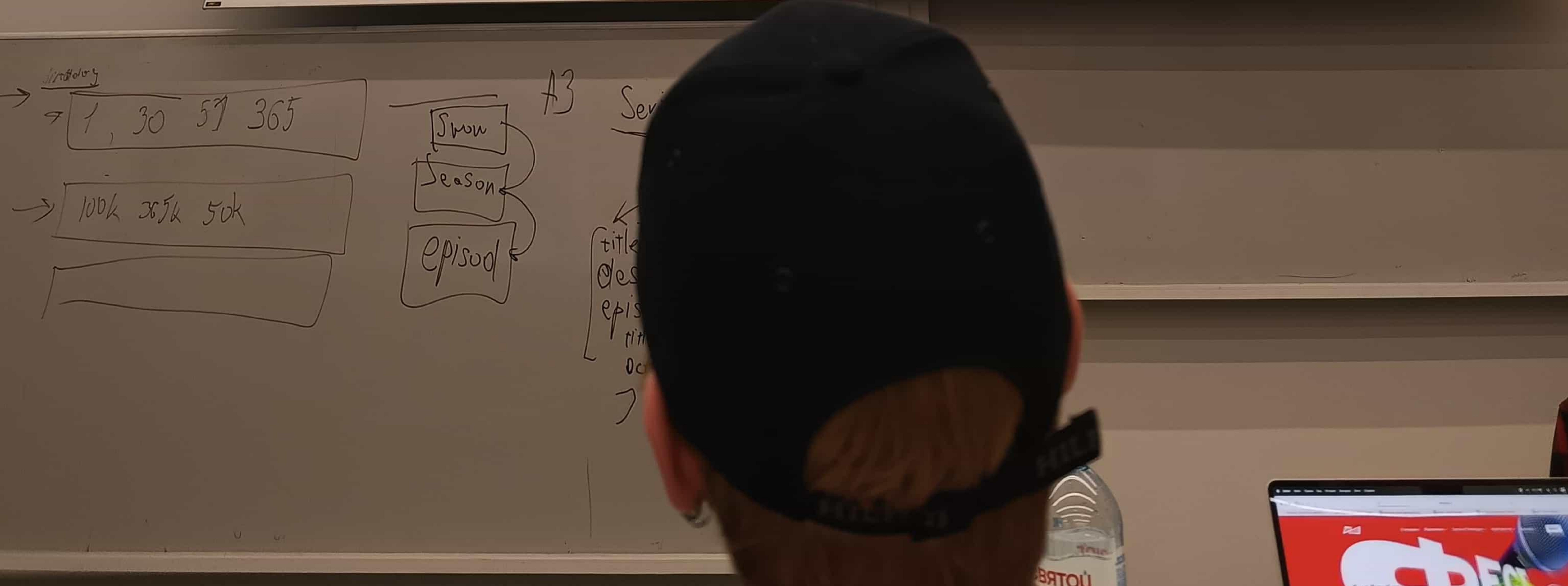
### Недостатки

- Ограничения целостности данных
- Сложность запросов
- Потенциальная избыточность



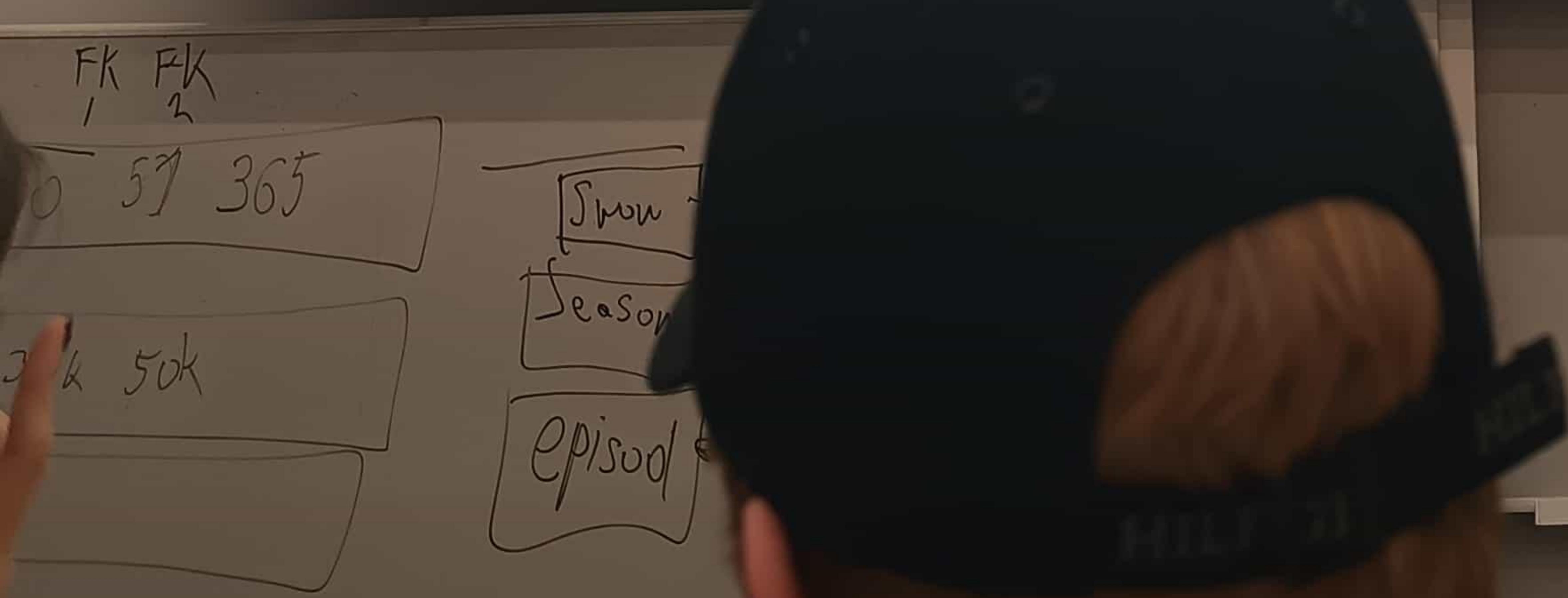
## Документоориентированная Применение

- Хранения логов
- OLTP-системы
- Разработка MVP
- Данные неструктурированы
- Обработка больших объемов данных



## Графовая Преимущества

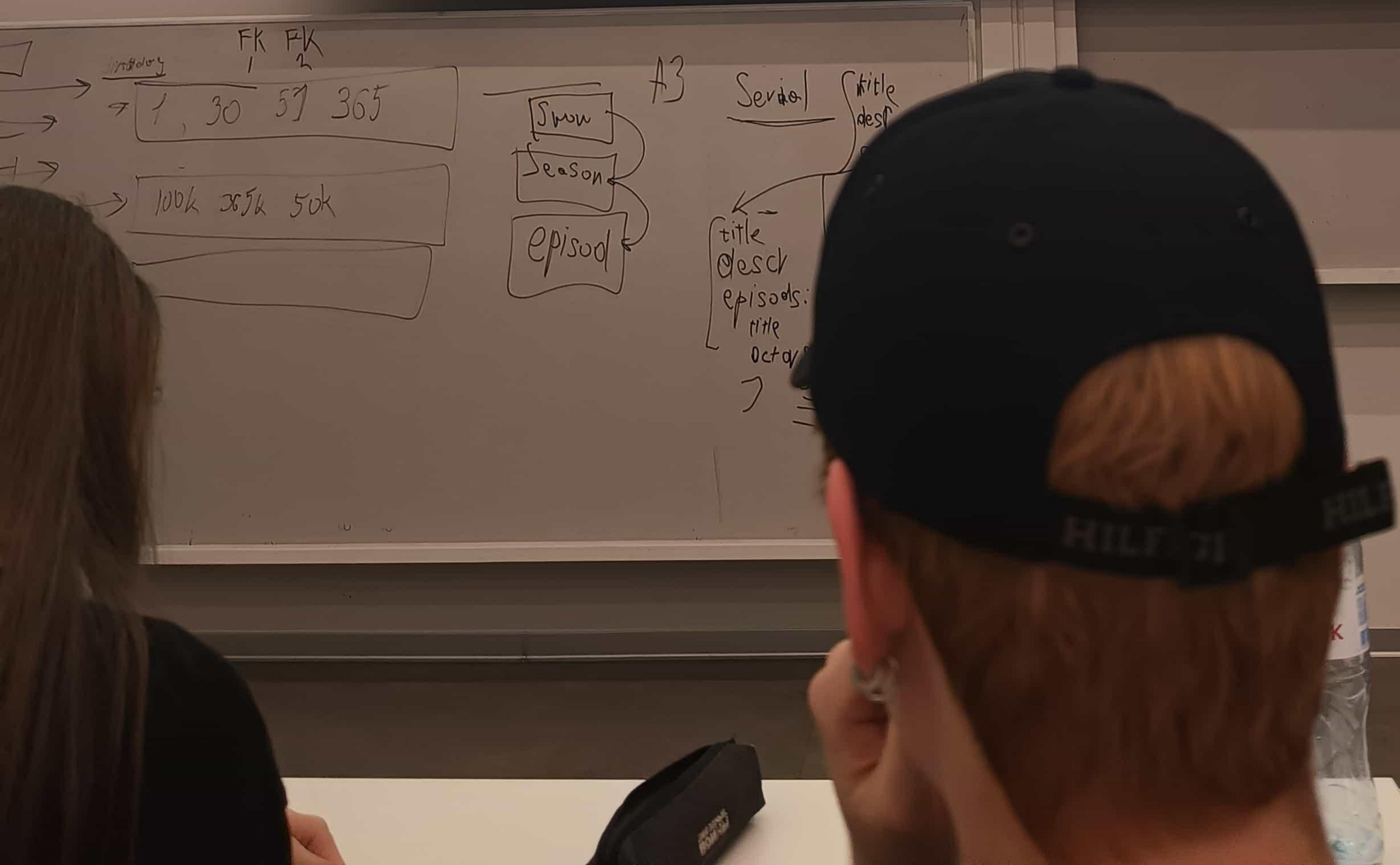
- Запросы, направленные на поиск путей, обход графа или анализ связей, выполняются очень быстро.
- Нет жестко фиксированной схемы
- Легко добавлять новые типы связей и узлов без необходимости пересмотра всей структуры данных



## Ключ-значение

### Особенности

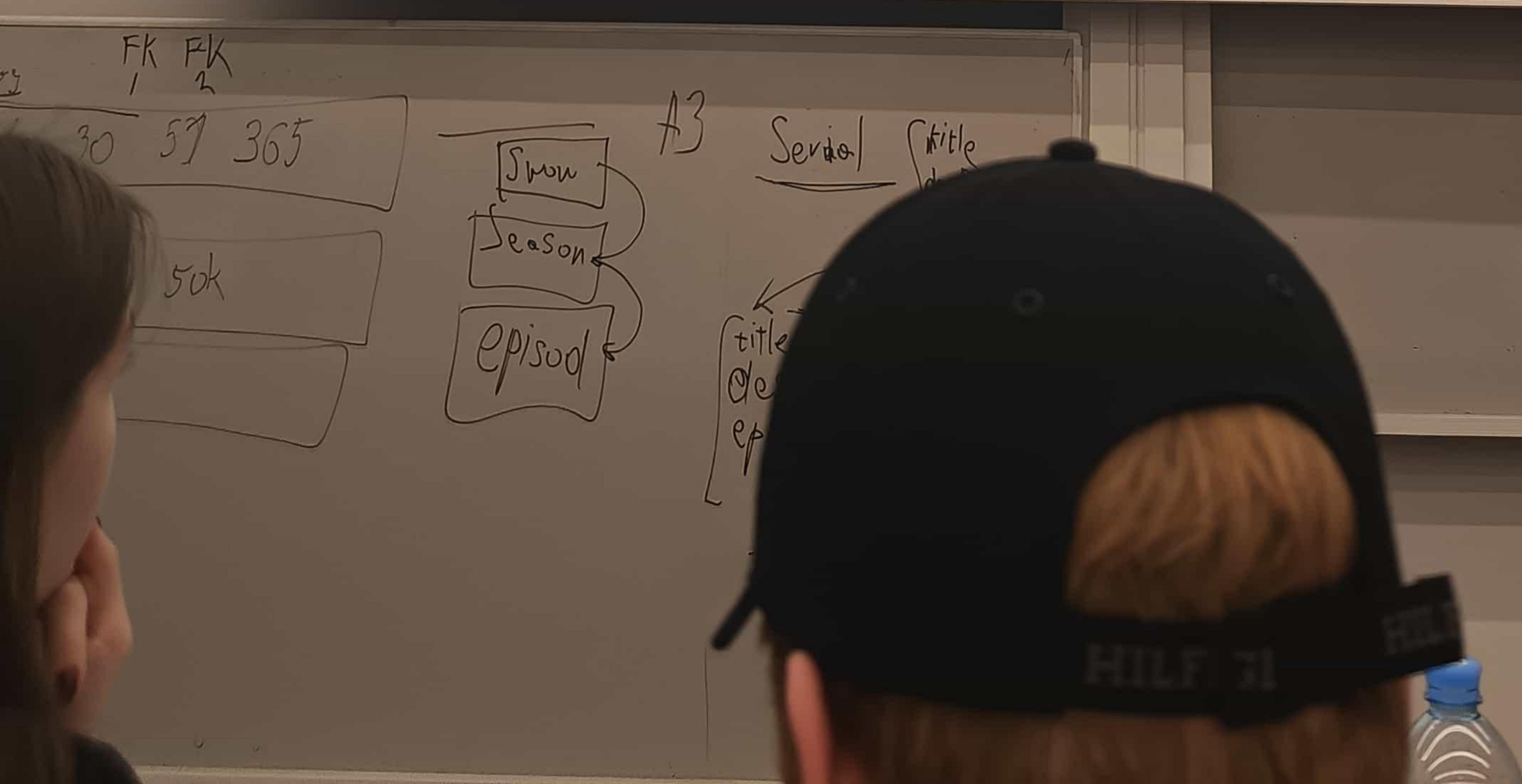
- Данные организованы в виде пар «ключ-значение»
- Значение может быть произвольного типа



## Ключ-значение

### Недостатки

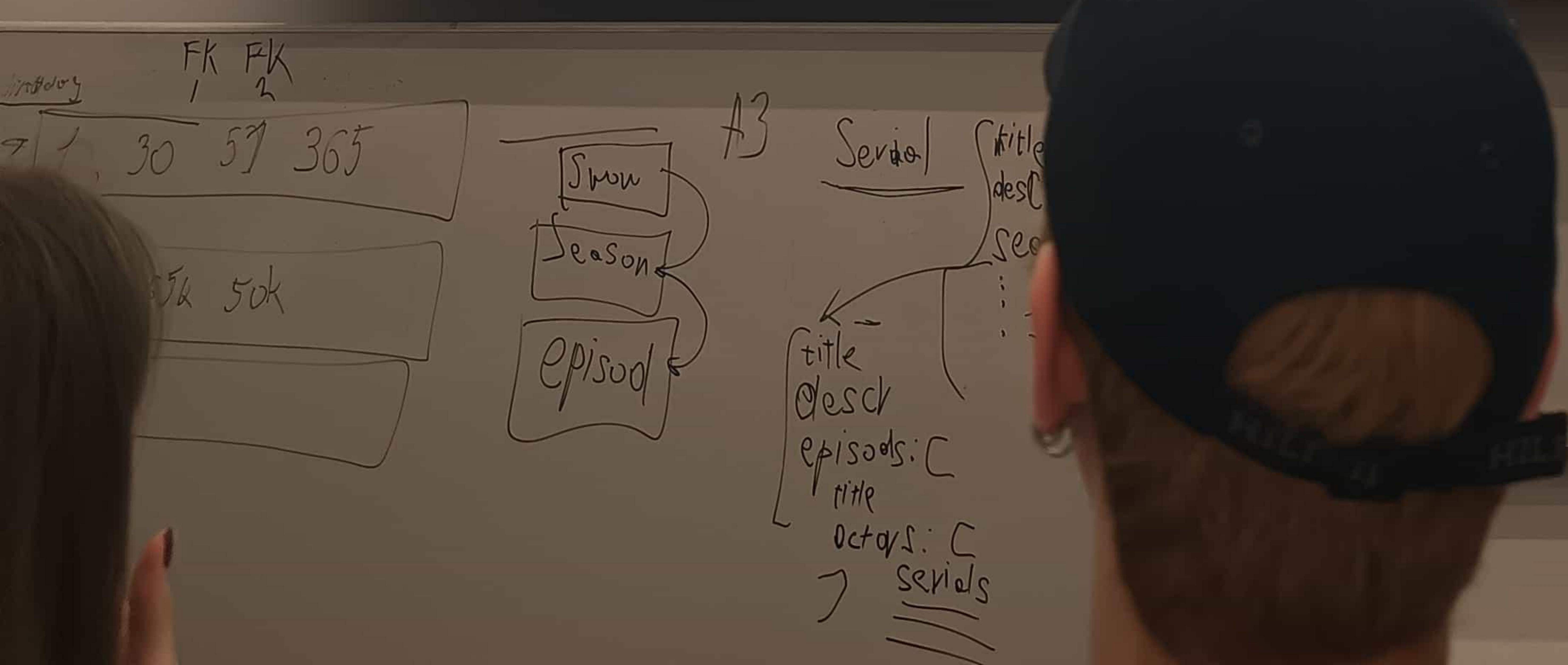
- Не подходят для сложных запросов, агрегаций или выборок по значениям внутри хранимых данных
- Не поддерживается сложное моделирование связей между данными



## Ключ-значение

### Применение

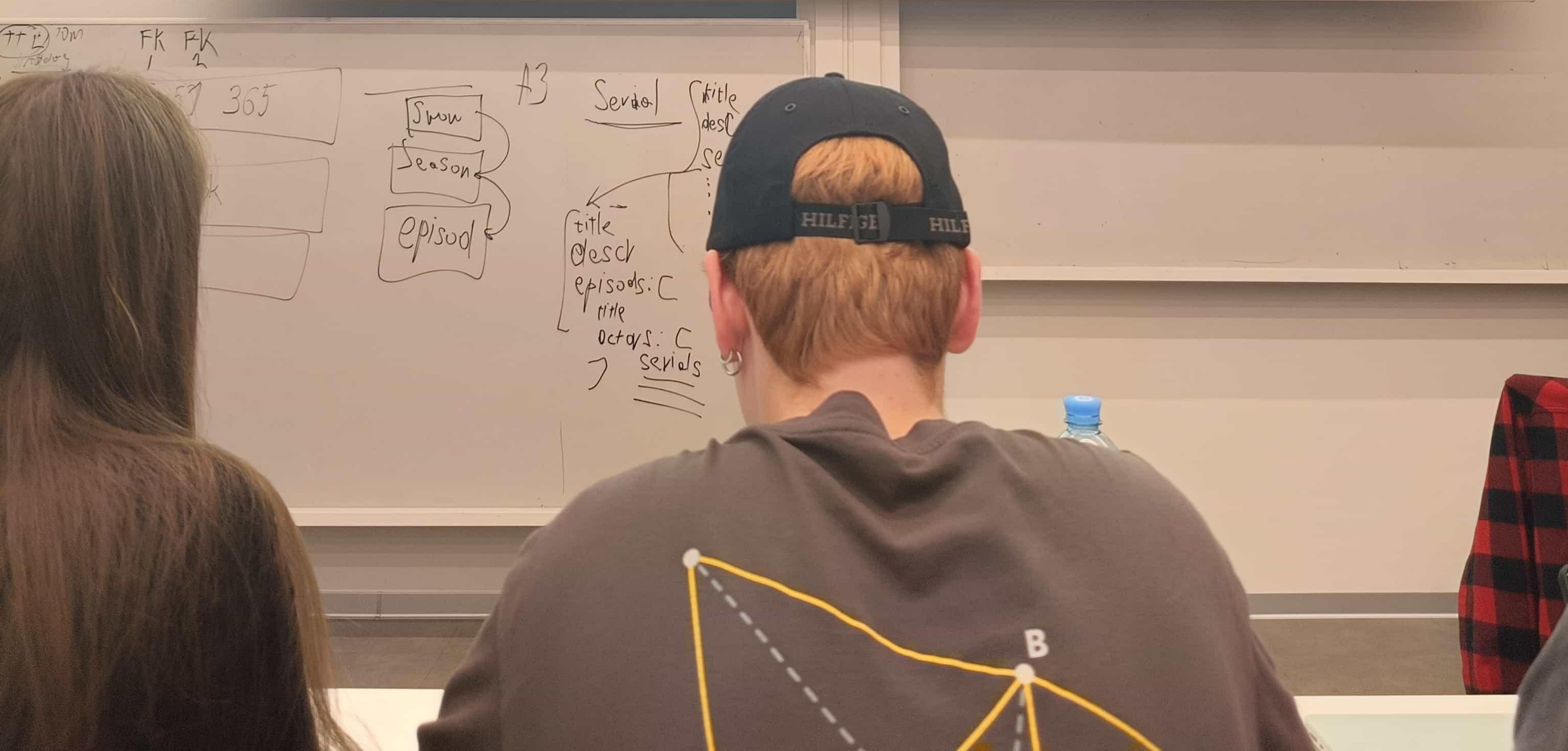
- Кэширование данных
- Хранение сессионных данных
- Реализация очередей
- Хранение конфигурационных данных



## Временные ряды

### Преимущества

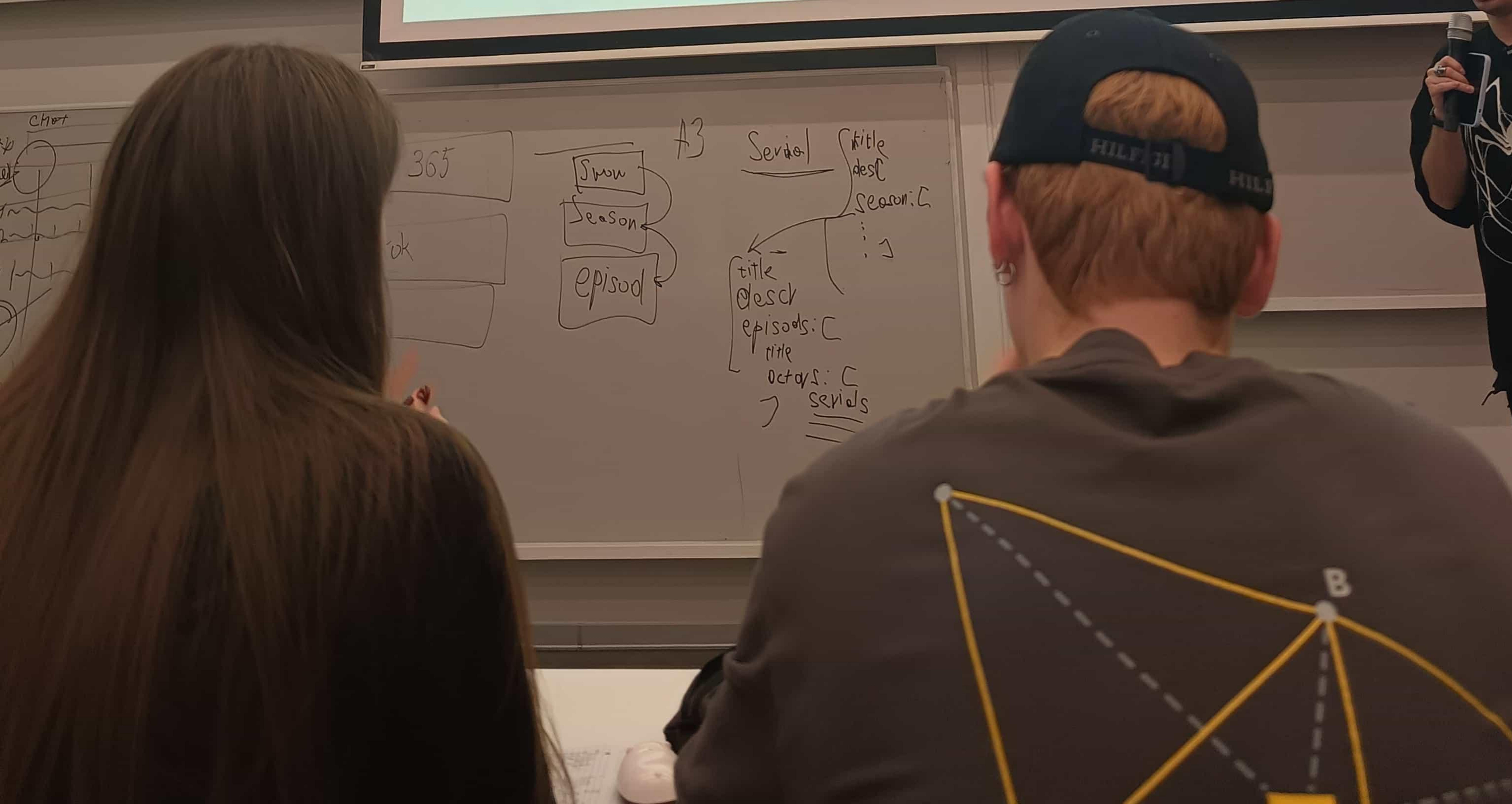
- Оптимизация на высокую скорость записи
- Позволяет легко делать агрегацию и интерполяция
- Хранение больших объемов данных
- Быстрый доступ к данным за определённые временные интервалы



# Временные ряды

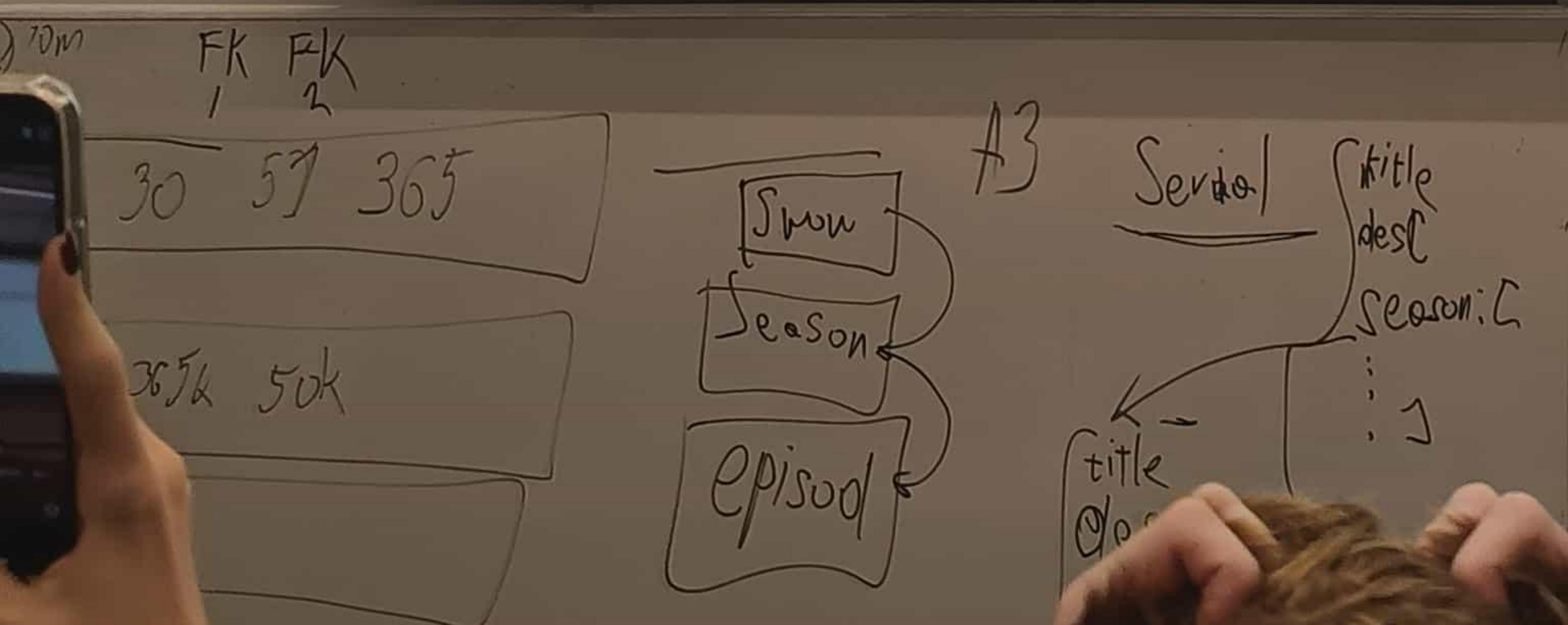
## Применение

- Мониторинг и логирование
- Финансовые рынки
- Интернет вещей



## Векторная Преимущества

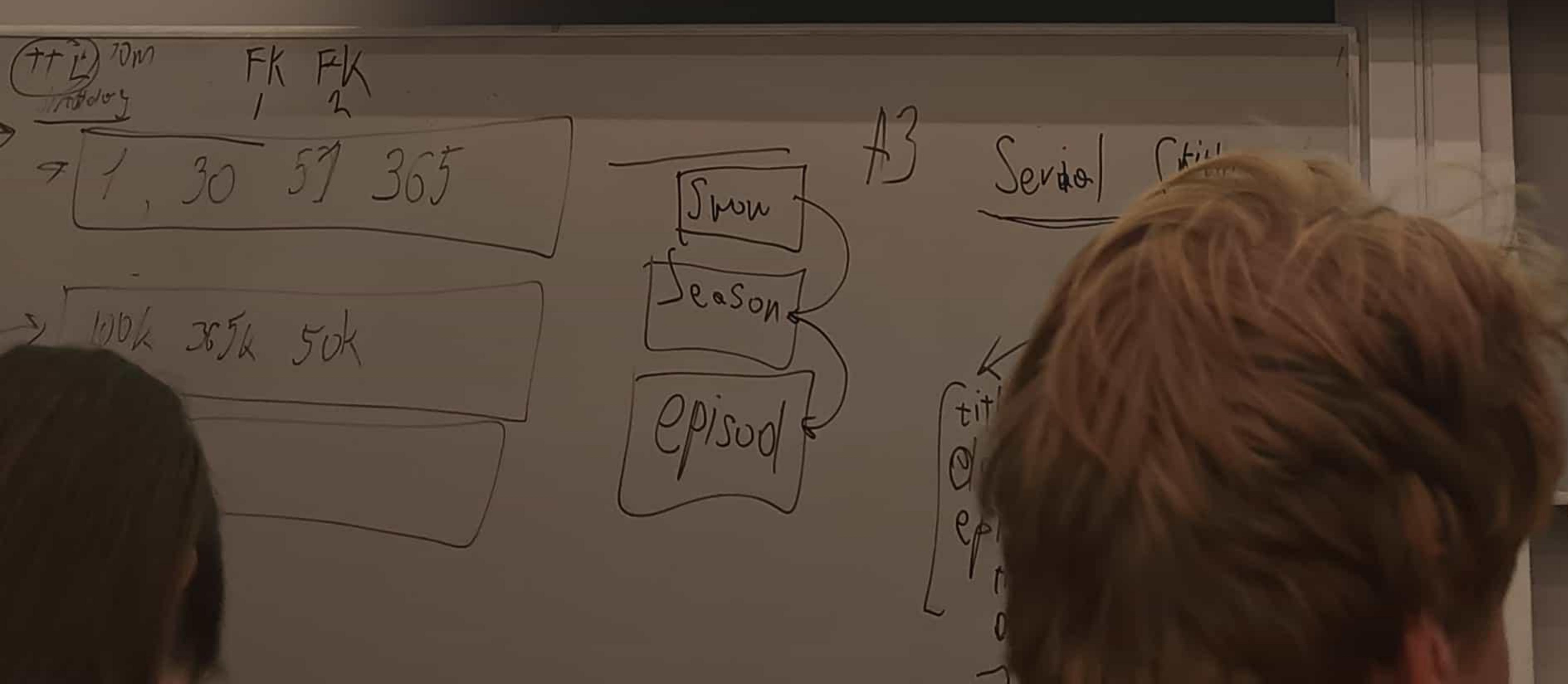
- Позволяют быстро выполнять запросы поиска похожих векторов благодаря специальным индексам и алгоритмам (напр. Hierarchical Navigable Small World)



## Хранилище данных

### Особенности

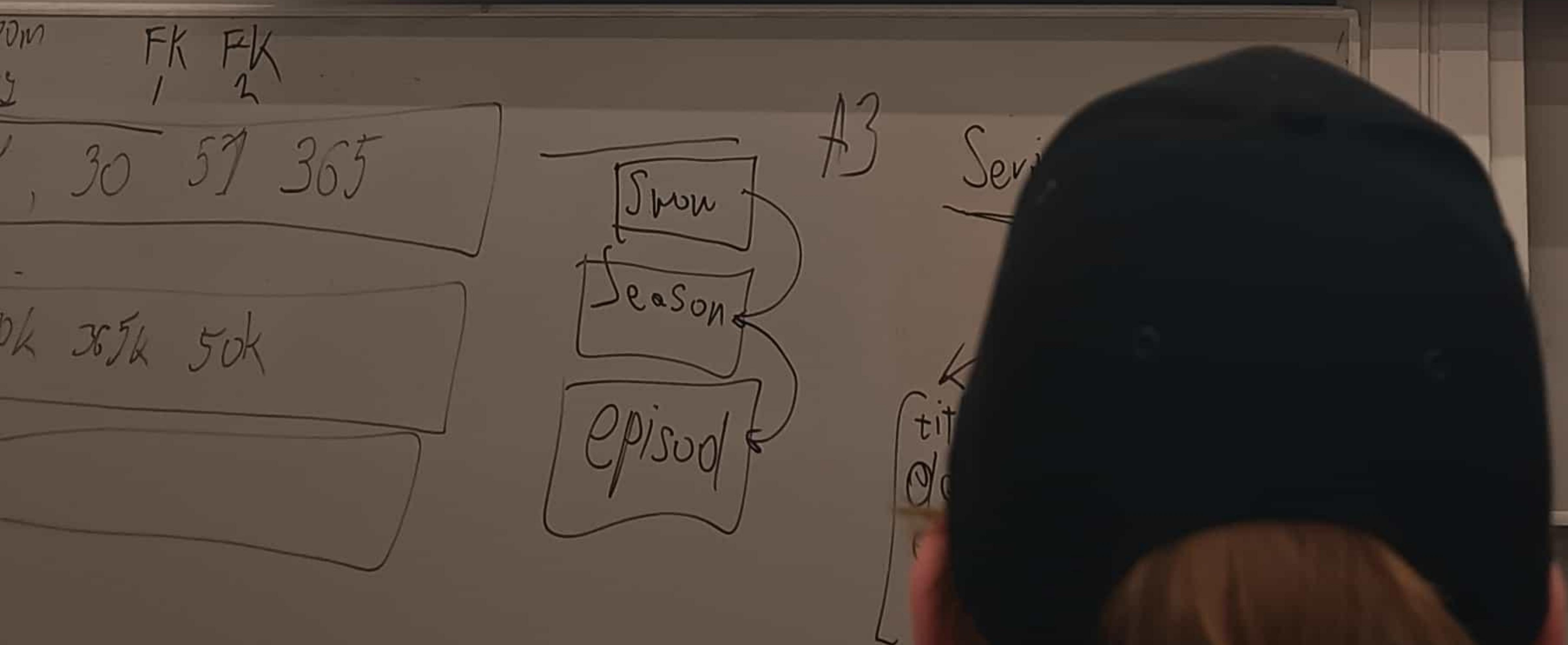
- предназначены для работы с файлами, такими как документы, изображения, видео, аудиофайлы.
- располагаются в каталогах и подкаталогах
- Хранит помимо файла - метаданные
- Доступ к файлам осуществляется по их именам и путям в файловой системе



## Хранилище данных

### Преимущества

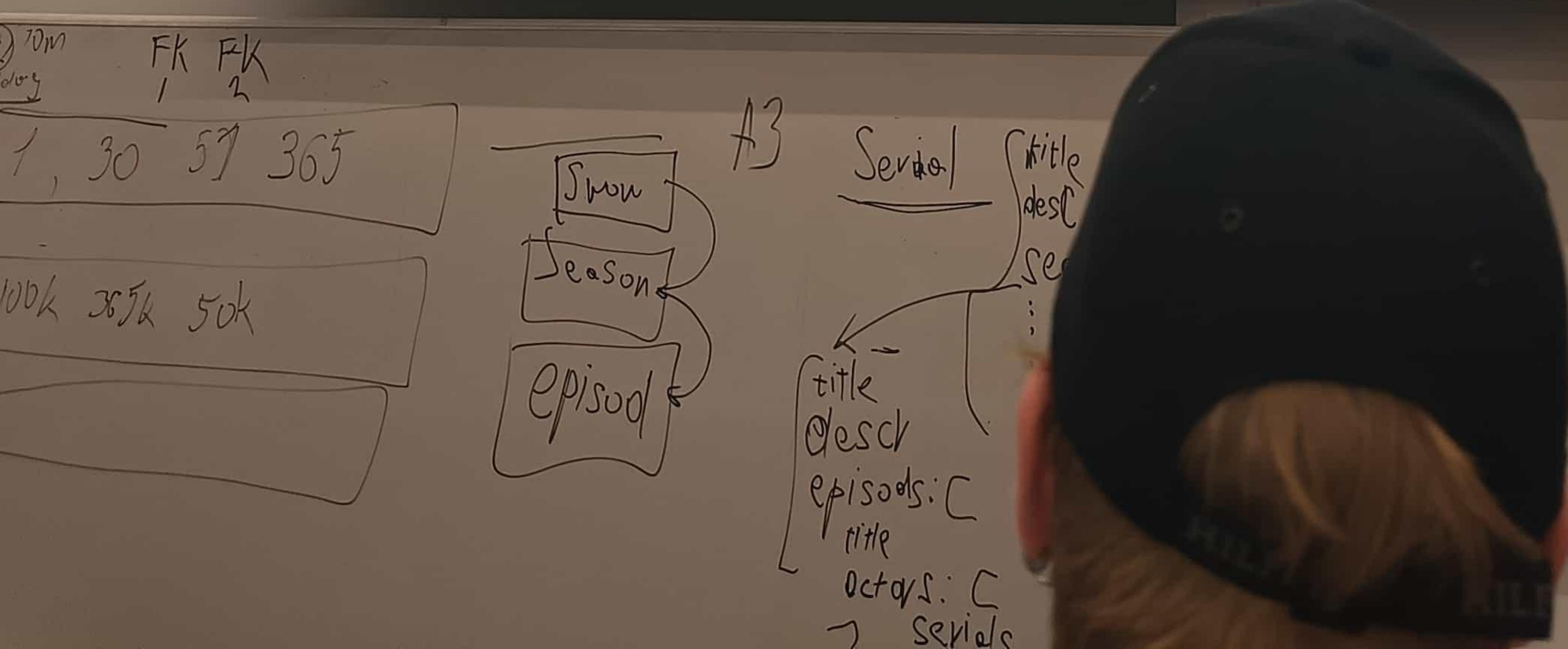
- Интеграция с операционными системами и стандартными программами для работы с файлами
- Позволяет выстраивать сложные модели доступа
- Позволяет разворачивать CDN (Content Delivery Network)



## Хранилище данных

### Недостатки

- Поиск данных осуществляется по именам файлов и метаданным, а не по содержимому



## Хранилище данных

### Применение

- Хранение слабоструктурированный информации (картинки, видео, документы, чертежи, аудио)

