**Исходные данные:**

E-commerce компания продает товары через онлайн-платформу.

Требования:

* Объем данных: компания собирает миллионы записей о продажах, клиентах и инвентаре ежедневно.
* Аналитика: необходимо обеспечить возможность проведения сложных аналитических запросов к данным для выявления тенденций продаж, сегментации клиентов и определения эффективности маркетинговых кампаний.
* Масштабируемость: компания растет, и вам нужна система, которая легко масштабируется с ростом объема данных и нагрузки.
* Реальное время: нужна возможность анализа данных в реальном времени для мониторинга заказов и складской информации.

**Задачи:**

1. Выберите одно или несколько хранилищ данных (реляционная база данных, NoSQL, колоночное хранилище и т. д.), которые, по вашему мнению, наилучшим образом соответствуют требованиям компании.
2. Обоснуйте свой выбор, предоставив аргументы, почему данное хранилище данных является оптимальным для конкретных требований.
3. Опишите структуру данных, которые будут храниться в выбранном хранилище (таблицы, колонки и связи).
4. Предложите схему интеграции данных из разных источников в хранилище.
5. Опишите, как бы вы настроили систему для обработки данных в реальном времени.

**Решение:**

1. NoSQL база данных наилучшим образом соответствует требованиям компании.

2. С учетом требований по работе с большим объемом данных, потребностью в проведении аналитики в режиме реального времени и возможности масштабирования, NoSQL база данных наилучшим образом соответствует требованиям компании, так как преимущества [[1]](#footnote-1) NoSQL баз данных по сути дублируют потребности заказчика:

*Гибкость схемы данных: NoSQL базы данных позволяют хранить данные без строгой схемы, что особенно полезно, когда формат данных может изменяться или когда необходима быстрая адаптация.*

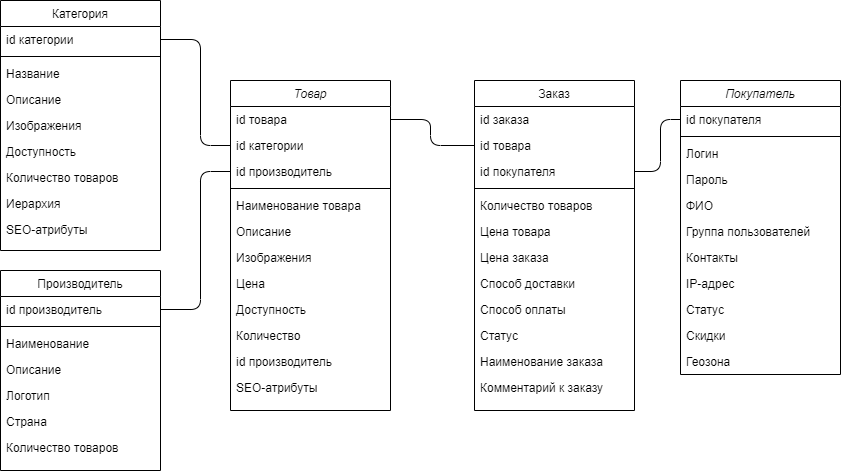
*Горизонтальное масштабирование: Многие NoSQL базы данных обеспечивают горизонтальное масштабирование, что позволяет обрабатывать высокий объем данных в режиме реального времени.*

*Высокая производительность: NoSQL базы данных, такие как Apache Cassandra или Amazon DynamoDB, спроектированы для обеспечения высокой производительности при записи и чтении данных, что важно для реального времени.*

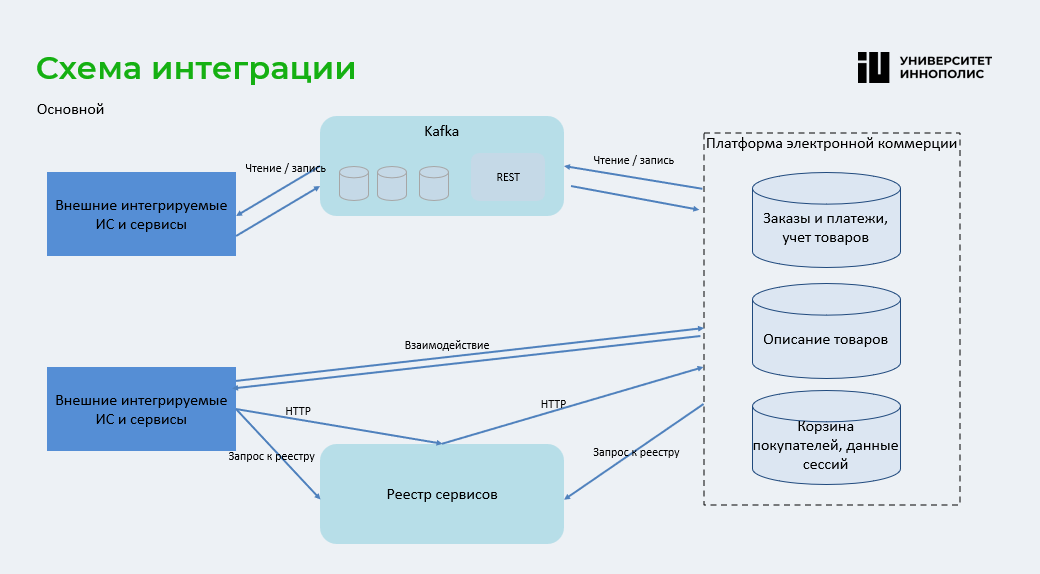
*Поддержка больших объемов данных: NoSQL базы данных могут легко обрабатывать большие объемы данных, что часто требуется при работе с потоками данных в реальном времени.*

*Гибридные решения: Некоторые NoSQL базы данных, например, Apache Cassandra, поддерживают гибридные модели данных, позволяя комбинировать SQL и NoSQL запросы для более сложных сценариев.*

3. Структура данных:



4. Для интеграции данных из разных источников в NoSQL хранилище предложила бы использовать API (позволяет обеспечивать обмен данными в режиме реального времени и по расписанию), а также использовать *Apache Kafka: распределенная потоковая платформа, которая может использоваться для интеграции нескольких источников данных и для потоковой передачи данных нескольким адресатам. Apache Kafka предоставляет надежный и масштабируемый способ передачи данных между различными компонентами вашего пайплайна обработки данных в реальном времени. Kafka особенно полезен при работе с большим объемом данных и требованиями к низкой задержке.*



1. Настройка системы для обработки данных в реальном времени:
   1. Пред настройка:
      1. железо и инфры (например, стоит убедиться в достаточности оперативной памяти и объеме дискового пространства).
   2. Настройка БД и потоков данных:
      1. Создание базы данных электронной коммерции, таблицы заказов и тд
      2. Создание необходимых категорий данных, в которых будут публиковаться и извлекаться данные
      3. Создание выделенного пользователя с доступом к таблице заказов
   3. Настройка обработчиков данных, синхронизация БД и инструментов интеграции:
      1. Настройка места источника данных
      2. Настройка места назначения данных
      3. Создание пайплайна
      4. Верификация данных
   4. Синхронизация данных
   5. Преобразование поступающих данных
   6. Обработка дубликатов записей с помощью аспертов
   7. Верификация процесса получения данных
   8. Написание запросов для анализа данных
   9. Обеспечиваем хранение данных
   10. Обеспечиваем мониторинг данных
   11. Настройка аналитических дашбордов, ориентированных на потребности конкретного пользователя

1. https://habr.com/ru/companies/otus/articles/764136/ [↑](#footnote-ref-1)