|  |  |  |
| --- | --- | --- |
| База данных | Плюсы | Минусы |
| Реляционная база данных | Имеют простую структуру, которая подходит к большинству типов данных.  Используют SQL, который широко распространен  Позволяют быстро обновлять данные, с помощью отношений, которые используются как указатели. Реляционные БД также поддерживают атомарные транзакции. | Поскольку каждый запрос выполняется к целой таблице, время выполнения запроса зависит от размера таблицы.  Масштабирование осуществляется добавлением вычислительных мощностей к компьютеру, на котором установлена БД.  Не поддерживаются объекты на основе ООП. |
| Документно-ориентированная база данных | Позволяют хранить объекты с разной структурой.  Могут отображать почти все структуры данных, включая объекты на основе ООП.  Поддерживают проверку схемы. Это значит, что вы можете сделать коллекцию со схемой.  Запросы к NoSQL очень быстрые.  В NoSQL масштабирование БД осуществляется добавлением компьютеров и распределением данных между ними, что позволяет автоматически добавлять ресурсы к БД, когда нам нужно, не провоцируя простои. | Обновление данных —медленный процесс в документной БД.  Атомарные транзакции по умолчанию не поддерживаются. |
| NoSQL база данных | Скорость обработки данных, масштабируемость, распределенность систем. | Небольшой функционал.  Недостаточная гибкость.  Немалое потребление ресурсов.  Необходимость специализированных знаний для работы с базой данных. |
| Ключ-значение | скорость работы;  простота модели хранения данных;  гибкость: значения могут быть любыми, включая JSON. | плохо масштабируются по мере усложнения моделей данных;  неэффективность при работе с группой записей;  отсутствие языка запросов. |
| Колоночные | Масштабируемость. Возможность хранения и изменения больших объемов информации.  Векторные аппаратные вычисления.  Возможность примерных вычислений.  Интеграция и совместимость с БД. Возможность работы по различным протоколам: НТТР АРI, wrарреr’s на Руthоn, РНР, NоdеJS, Реrl, Rubу.  Оптимизация под HDD-носители. Возможна работа с внешними хранилищами.  Интеграция под пользовательские онлайн-запросы. | Отсутствие транзакций.  Сложность «точечной» работы. Система управления базами данных работает с крупными массивами. В ней отсутствуют удобные инструменты для поиска, обновления или удаления конкретной строки.  Высокие требования к ОЗУ сервера.  Возможность возникновения багов.  Уязвимость HTTP-протокола. |
| Графовые | Простая и понятная модель данных  Нет необходимости составлять схему данных  Простой и понятный язык запросов, а также быстрое получение результатов по сравнению с реляционными базами данных | низкую скорость доступа к данным, плохую поддержку неструктурированных данных, сложность масштабирования и образование большого количества таблиц |

В данном случае для хранения основной информации будем использовать реляционную базу данных. Ее основные преимуществами для данного кейса являются: надежность и согласованность данных, подходит для структурированных данных и сложных вопросов, поддерживает транзакции и удобно работать с отношениями между таблицами. Для хранения дополнительной информации о товарах будем использовать документно-ориентированную базу данных. Ее основными преимуществами для данного кейса являются: гибкость в работе с полу-структурированными данными, хорошо масштабируется.

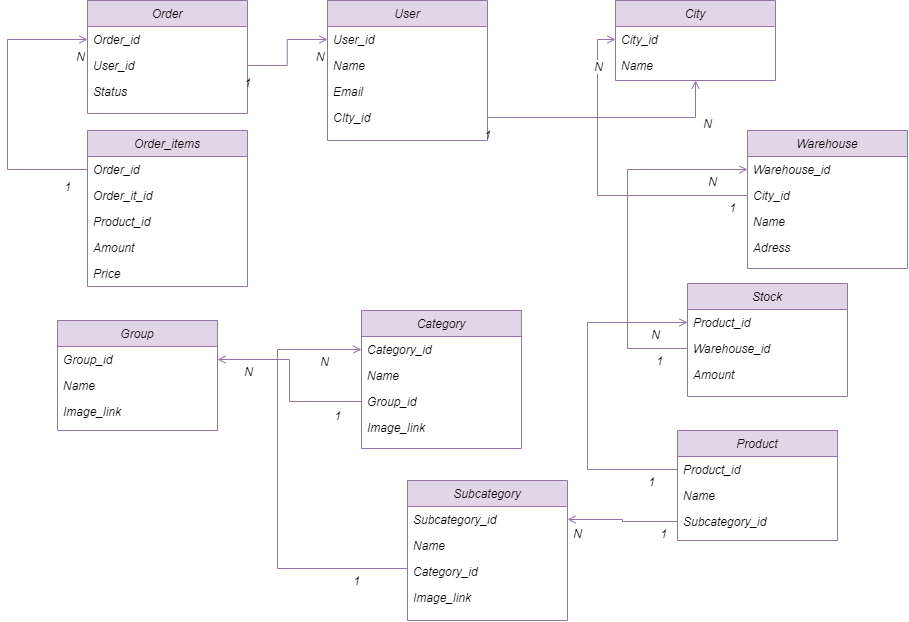


Схема реляционной БД

В документно-ориентированной базе данных будет располагаться таблица «Product\_Info». Обязательными полями является «Product\_id» и «Image\_link». Дополнительные поля, являются специфичными для каждого вида товара.

Медиаконтент будем хранить на облачном сервере, а в базу данных будем заносить ссылки на файлы.

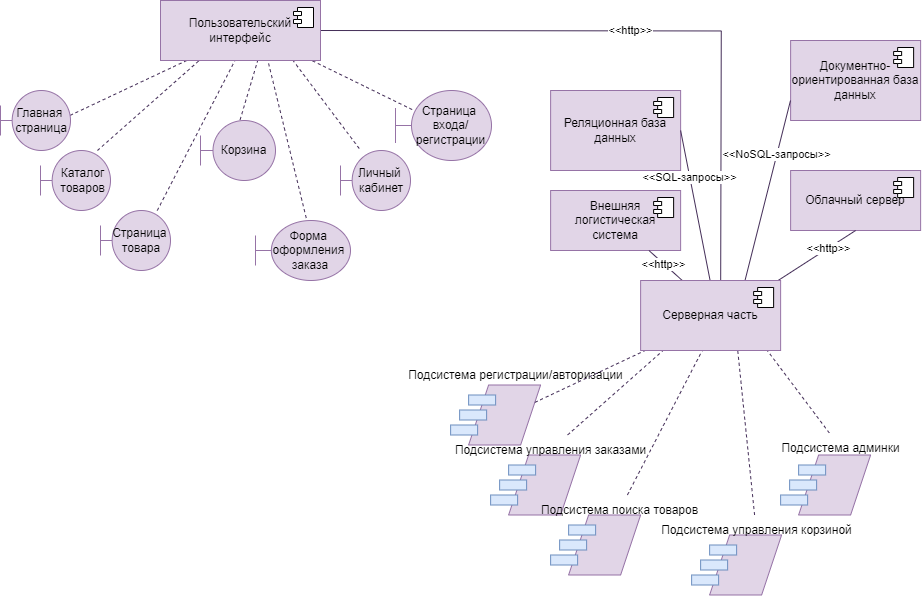


Схема взаимодействия компонентов системы