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.***

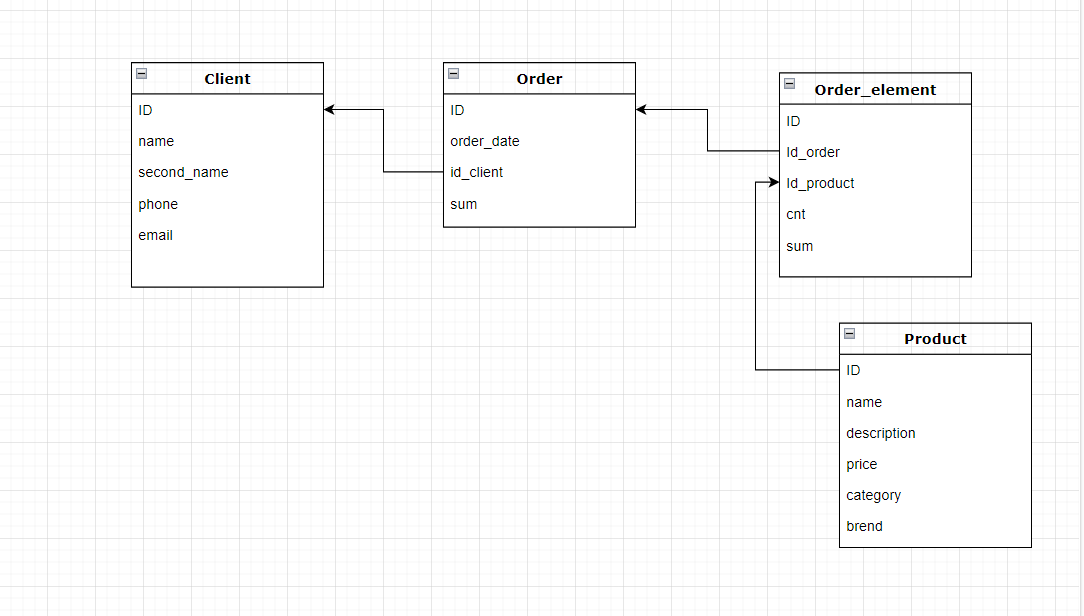
Поскольку Реляционные БД лучше обрабатывают небольшие транзакции с большой долей вставок, можно использовать PostgreSQL. Но при больших объемах логичнее использовать Колоночные БД. Они позволяют ускорить чтение при больших (сто миллионов записей и больше) объемах. Предлагаю остановить выбор на Clickhouse

1. ***Обоснуйте свой выбор, предоставив аргументы, почему данное хранилище данных является оптимальным для конкретных требований.***

ClickHouse, как и другие реляционные системы, использует синтаксис SQL, его диалект. Поэтому разработчикам разобраться с CLickHouse будет нетрудно.

Также если нужно читать по big data, то в строковой системе, например, PostgreSQL, строки всегда читаются полностью. Только проходя через колонку, можно определить ее имя и необходимость для текущей задачи. Пропустить колонки, которые не нужны для анализа, невозможно. В ClickHouse проходить все строки не обязательно: она использует только те колонки, которые участвуют в запросе. Каждая колонка столбцовой системы хранится в отдельном файле, а в строковой вся таблица распределяется по 2 — 3 файлам. Поэтому обратиться к конкретному столбцу ClickHouse может намного быстрее.Также благодаря хранению каждой колонки данных в отдельном файле базу проще сжимать. В обычных строковых СУБД файлы содержат в себе разные типы данных, и сжать их оптимально не выйдет. При этом чем меньше размер базы данных, тем быстрее происходит поиск, запрос требует меньше операций ввода-вывода.

1. ***Опишите структуру данных, которые будут храниться в выбранном хранилище (таблицы, колонки и связи)***

******

***Таблица продуктов***

Таблица продуктов должна содержать информацию о продуктах, которые продает интернет-магазин. В таблице должны быть следующие поля:

— ID продукта – уникальный идентификатор продукта;

— Название продукта – название продукта;

— Описание продукта – описание продукта;

— Цена – стоимость продукта;

— Категория – категория продукта;

— Бренд – бренд продукта.

***Таблицы клиентов***

Таблица клиентов должна содержать информацию о клиентах, которые делают заказы в интернет-магазине. В таблице должны быть следующие поля:

— ID клиента – уникальный идентификатор клиента;

— Имя – имя клиента;

— Фамилия – фамилия клиента;

— Телефон – номер телефона клиента;

— Email – электронный адрес клиента.

***Таблица заказов***

Таблица заказов должна содержать информацию о заказах, сделанных клиентами в интернет-магазине. В таблице должны быть следующие поля:

— ID заказа – уникальный идентификатор заказа;

— Дата заказа – дата, когда был сделан заказ;

— ID клиента – уникальный идентификатор клиента, сделавшего заказ;

— Сумма заказа – стоимость заказа.

***Таблица элементов заказа***

Таблица элементов заказа должна содержать информацию о каждом элементе заказа. В таблице должны быть следующие поля:

— ID элемента заказа – уникальный идентификатор элемента заказа;

— ID заказа – уникальный идентификатор заказа, содержащего элемент заказа;

— Количество – количество товаров, заказанных клиентом;

— Цена – стоимость товаров, заказанных клиентом.

1. ***Предложите схему интеграции данных из разных источников в хранилище.***

Для обеспечения отказоустойчивости и масштабируемости ClickHouse разворачивается в кластерном режиме. Для координации процесса репликации используется ZooKeeper . Zookeeper выполняет роль координации распределенных сервисов, не обеспечивая передачу данных из одного источника в другой. Kafka позволяет публиковать потоки данных и подписываться на них, организовать отказоустойчивое хранилище и обрабатывать потоки по мере их появления через использование таблиц с указанием специальных параметров.

1. ***Опишите, как бы вы настроили систему для обработки данных в реальном времени.***

Можно использовать материализованное представление таблицы Kafka в ClickHouse. Она позволяет СУБД непрерывно получать сообщения от Кафка в фоновом режиме.

Поток данных реализуется через потребителя Kafka, созданного с помощью движка;

материализованное представление преобразует данные от движка и помещает их в заранее созданную таблицу с нужной структурой;

Одна таблица Кафка в Кликхаус может иметь любое число материализованных представлений, поскольку они не считывают данные непосредственно, а блоками получают новые записи. Материализованные представления в ClickHouse не читают Kafka-таблицы, а получают данные от них при срабатывании триггера на INSERT. Поэтому одну Kafka-таблицу можно использовать как для сырых запросов, так и для агрегированных и сгруппированных данных по ним. Благодаря этому данные можно записать в несколько таблиц с разным уровнем детализации.