# **1. Introduction**

## **1. Может ли быть в таблице реляционной базы данных несколько строк с повторяющимся PK? Почему?**

Нет, в реляционной базе данных ключевое поле, которое определено как Primary Key (PK), должно быть уникальным для каждой записи в таблице. Одной из основных целей Primary Key является обеспечение уникальности идентификации каждой строки в таблице, и, следовательно, оно не может содержать дублирующихся значений.

Если бы Primary Key мог быть неуникальным, то это привело бы к неоднозначности идентификации записей в таблице, что нарушило бы основные принципы целостности данных в реляционной модели.

Поэтому Primary Key всегда должен быть уникальным и обеспечивать уникальность идентификации записей в таблице.

## **2. К какому типу относится субд Redis? В чем особенность данной субд при работе с памятью?**

СУБД Redis относится к типу NoSQL (нереляционных баз данных) и более конкретно, к типу ключ-значение (key-value). Особенность Redis заключается в том, что она является in-memory базой данных, что означает, что данные хранятся в оперативной памяти, а не на диске. Это делает Redis очень быстрой и подходящей для кэширования данных, обработки сессий и других приложений, где скорость доступа к данным критически важна.

## **3. Что является аналогом таблицы (в реляционной СУБД) в MongoDB?**

## **Что является аналогом строки/записи(в реляционной СУБД) в MongoDB? Приведите небольшой пример того, как может выглядеть данный аналог.**

В MongoDB, аналогом таблицы в реляционной СУБД является коллекция (collection). Коллекция в MongoDB может содержать документы, и она служит для группировки связанных документов. Документы в коллекции могут иметь разную структуру, что отличает MongoDB от реляционных СУБД, где структура таблицы фиксирована.

Аналогом строки или записи (в реляционной СУБД) в MongoDB является документ. Документ представляет собой JSON-подобный объект, который может содержать разные поля и значения. Документы в коллекции могут иметь различные наборы полей. Вот пример аналога строки/записи в MongoDB:

// Пример документа в коллекции "users" (аналог строки)

{

"\_id": ObjectId("5f6a7d7a82b7a02b9c5606f3"),

"name": "John Smith",

"email": "john@example.com",

"age": 30

}

## **4. Выберите наиболее подходящие СУБД для следующих задач:**

**-Анализ поведения знаменитостей в социальной сети:**

В данной задаче для анализа связей между пользователями и их действиями, наиболее подходящей СУБД может быть графовая база данных, такая как Neo4j. Графовые СУБД хорошо подходят для моделирования и анализа связей, что делает их идеальным выбором для социальных сетей и сценариев с анализом графовых данных.

**-Онлайн-аналитика действий пользователей на сайте онлайн магазина:**

В данной задаче для аналитики действий пользователей на сайте и управления информацией о клиентах, товарах и заказах наиболее подходят реляционные СУБД, такие как PostgreSQL, MySQL или Microsoft SQL Server. Реляционные базы данных обеспечивают структурированное хранение данных и поддерживают SQL-запросы, что полезно для анализа и отчетности.

**-Анализ документации по фильмам и сериалам:**

Для анализа и хранения документации с разнообразной структурой, такой как название, год выпуска, список актеров и жанр, хорошо подходят документоориентированные СУБД, такие как MongoDB или Elasticsearch. Эти СУБД позволяют хранить и анализировать данные с гибкой структурой.

**-Проектирование наиболее оптимальных маршрутов поставок для маркетплейса:**

Для проектирования оптимальных маршрутов поставок, которые могут включать геоданные и графовые связи, наиболее подходящей СУБД может быть геоинформационная база данных, такая как PostGIS (расширение для PostgreSQL), которая поддерживает геопространственные данные и анализ.

**-Хранение кэшированных HTML страниц пользователей, посещающих сайт:**

Для хранения кэшированных данных, таких как HTML страницы, наиболее подходят ключ-значение (key-value) СУБД, и здесь Redis может быть хорошим выбором. Redis быстро хранит данные в памяти и поддерживает операции чтения и записи с низкой задержкой, что полезно для кэширования.