1. In-memory Database (БД в памяти) - это тип базы данных, в котором данные хранятся и обрабатываются непосредственно в оперативной памяти компьютера, в отличие от традиционных баз данных, где данные обычно хранятся на постоянном носителе, таком как жесткий диск. В БД в памяти данные доступны намного быстрее, так как операции чтения и записи выполняются непосредственно в памяти, а не требуют обращения к диску. Это обеспечивает высокую производительность и низкую задержку при обработке запросов.
2. Хэш-таблица (Hash Table) - это структура данных, которая использует хэш-функцию для преобразования ключей в индексы массива. Хэш-таблица обеспечивает эффективный доступ к данным и быстрое выполнение операций вставки, удаления и поиска. Она состоит из массива бакетов (слотов), где каждый слот может хранить одно или несколько значений данных. При вставке элемента в хэш-таблицу, ключ элемента подвергается хэшированию, и полученный хэш-значение используется для определения индекса массива, в котором будет храниться элемент. При поиске или удалении элемента происходит вычисление хэш-значения ключа и поиск элемента в соответствующем слоте.
3. В Redis персистентность данных обеспечивается через комбинацию нескольких механизмов. По умолчанию Redis сохраняет данные на диск, используя механизм снапшотов (RDB - Redis Database) и журналов транзакций (AOF - Append-Only File).

Механизм снапшотов (RDB) позволяет создавать точные копии данных Redis в определенные моменты времени. Снапшоты создаются асинхронно, и Redis периодически сохраняет данные на диск в компактном двоичном формате. При перезапуске Redis загружает последний снапшот с диска и восстанавливает данные.

Журналы транзакций (AOF) записывают все операции записи (вставка, обновление, удаление) в простой текстовый файл. При перезапуске Redis выполняет операции восстановления, применяя записи из журнала транзакций в порядке их выполнения, чтобы восстановить состояние базы данных.

Redis также предлагает комбинированный режим, в котором используются и снапшоты (RDB), и журналы транзакций (AOF). В этом режиме Redis восстанавливает данные из AOF-журнала, если он существует, иначе использует снапшот.

1. Команда SET в Redis используется для установки значения по ключу. Она принимает два аргумента: ключ и значение, и устанавливает значение ключа в указанное значение.

Команда GET используется для получения значения по ключу. Она принимает ключ в качестве аргумента и возвращает значение, связанное с этим ключом.

Команда DEL используется для удаления ключа и связанного с ним значения. Она принимает один или несколько аргументов и удаляет все указанные ключи из базы данных Redis.

Команда GETSET используется для полученияи одновременной установки нового значения для указанного ключа. Она принимает два аргумента: ключ и новое значение. Команда GETSET сначала возвращает текущее значение ключа, а затем устанавливает новое значение.

1. Команды INCR, DECR, INCRBY и DECRBY используются для работы с числовыми значениями, хранящимися в Redis:

* Команда INCR (increment) увеличивает значение ключа на 1. Если ключ не существует, он создается с начальным значением 0 перед выполнением операции.
* Команда DECR (decrement) уменьшает значение ключа на 1. Если ключ не существует, он создается с начальным значением 0 перед выполнением операции.
* Команда INCRBY увеличивает значение ключа на указанную величину (целое число). Если ключ не существует, он создается с начальным значением 0 перед выполнением операции.
* Команда DECRBY уменьшает значение ключа на указанную величину (целое число). Если ключ не существует, он создается с начальным значением 0 перед выполнением операции.

1. Команды MSET и MGET используются для работы с несколькими ключами и их значениями в Redis:

* Команда MSET (multi-set) принимает несколько аргументов в формате "ключ значение ключ значение" и устанавливает значения для соответствующих ключей.
* Команда MGET (multi-get) принимает несколько аргументов в формате "ключ ключ ключ" и возвращает значения для указанных ключей.

1. Команды HSET и HGET используются для работы с хэш-таблицами (хеш-мапами) в Redis:

* Команда HSET (hash set) принимает три аргумента: имя хэша (ключ), поле и значение. Она устанавливает значение поля в указанном хэше.
* Команда HGET (hash get) принимает два аргумента: имя хэша (ключ) и поле. Она возвращает значение поля из указанного хэша.

1. Команды HMSET и HMGET используются для работы с хэш-таблицами (хеш-мапами) в Redis:

* Команда HMSET (hash multi-set) принимает три или более аргументов: имя хэша (ключ), поля и значения. Она устанавливает значения нескольких полей в указанном хэше.
* Команда HMGET (hash multi-get) принимает два или более аргумента: имя хэша (ключ) и поля. Она возвращает значения полей из указанного хэша.

1. Команда EXISTS используется для проверки наличия ключа в Redis. Она принимает один аргумент - ключ и возвращает 1, если ключ существует, и 0, если ключ не существует.
2. Команда MULTI используется для запуска транзакции в Redis. Она указывает Redis начать мультикомандный блок, в котором последующие команды будут рассматриваться как атомарные операции, выполняемые последовательно. Команды внутри блока транзакции выполняются только после вызова команды EXEC, которая фактически выполняет все команды в блоке.
3. Метод sendCommand в библиотеке Redis используется для отправки произвольных команд на Redis-сервер. Он позволяет отправить команду Redis, указывая ее имя и аргументы в качестве параметров метода sendCommand. Этот метод предоставляет более гибкий способ отправки команд, чем использование предопределенных методов, таких как SET, GET и т. д.