1. Поясните понятие **in-memory Database**.

База данных в памяти (англ. **In**-**memory** **database**, IMDB) — **это** система управления базами данных, которая при хранении компьютерных данных в основном полагается на оперативную память.

1. Поясните понятие **хэш-таблица**.

**Хеш-табли́ца** — это [структура данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), реализующая интерфейс [ассоциативного массива](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2), а именно, она позволяет хранить пары (ключ, значение) и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу.

1. Каким образом обеспечивается персистентность данных в СУБД Redis?

**Персистентные** структуры **данных** (англ. persistent data structure) — **это** структуры **данных**, которые при внесении в них каких-то изменений сохраняют все свои предыдущие состояния и доступ к этим состояниям.

В Redis реализовано два persistence-режима: RDB и AOF.

При включении **RDB** система создает компактные снэпшоты данных в заданные интервалы времени. Это хороший подход для восстановления информации в случае сбоя. Бэкапы пишутся в параллельном процессе при помощи команды **fork()**, который в случае большой БД затратен по ресурсам и времени. Кроме этого, в случае неожиданного выключения сервера, все данные, которые не были записаны или находились в процессе создания резервной копии, будут утеряны.

**AOF** представляет собой лог операций, которые выполняют клиенты. Метод расшифровывается как Append Only File, то есть все новые данные добавляются к уже имеющимся данным, причем, каждую секунду по умолчания. Так что в случае сбоя потери будут минимальны. Но подход немного медленнее RDB, лог-файл существенно больше, производительность зависит от параметров fsync.

Хотите максимальной производительности — отключите persistence полностью. Или же настройте частоту создания снэпшотов. По умолчанию установлены параметры:

* Создать копию, если было хотя бы одно изменение в течение 15 минут (900 секунд);
* Создать копию, если в течение 5 минут (300 секунд) было хотя бы 10 изменений;
* Создать копию, если за минуту было 10 000 изменений.

1. Поясните назначение команд СУБД Redis: **set**, **get**, **del, getset**.

**GET ключ** — Возвращает значение ключа.

**GETSET ключ значение** — Устанавливает в переданный ключ строковое значение и возвращает предыдущее.

1. Поясните назначение команд СУБД Redis: **incr**, **decr**.

**INCR ключ** — Увеличивает на единицу значение числа. В случае, если заданный ключ содержит строку, будет вызвана ошибка.

**DECR ключ** — Уменьшает на единицу значение числа. В случае, если заданный ключ будет содержать строку, будет сгенерирована ошибка.

1. Поясните назначение команд СУБД Redis: **mset, mget**.

**MGET ключ [ключ …]** — Возвращает значение ключа / ключей, переданных в параметрах.

**MSET ключ значение [ключ значение …]** — Устанавливает значение ключа / значения ключей, которые переданы в параметрах.

1. Поясните назначение команд СУБД Redis: **hset, hget.**

**HGET ключ поле значение** — Возвращает значение, которое ассоциировано с полем в хэше

**HSET ключ поле значение** — Добавляет в хэш поле и значение. Если такого ключа не существовало, он будет добавлен. В случае, если такое поле в хэше уже существует, оно будет перезаписано.

1. Поясните назначение команд СУБД Redis: **hmset,hmget.**

**HMGET ключ поле [поле …]** — Получает значение поля / полей указанного хэша.

**HMSET ключ поле значение [поле значение …]** — Записывает значения в поля хэша.

1. Поясните назначение команды СУБД Redis: **exists**.

Проверяет – есть ли поле с таким ключом