# Ответьте на следующие вопросы:

1. **Что такое последовательность?**

Последовательность — это объект базы данных, предназначенный для генерации числовой последовательности. Часто используется для генерации значений айдишников для первичных ключей, снимая, во-первых, с программиста обязанность это вручную прописывать в коде, а во-вторых, снижая нагрузку с БД. Если бы пользователи вынуждены были создавать последовательные числа программно, то Oracle пришлось бы постоянно блокировать и разблокировать запись, хранящую максимальное значение таких последовательностей, чтобы гарантировать упорядоченное их возрастание.

1. **Перечислите основные параметры последовательности.**

* START WITH – число, с которого начнется отсчет. По дефолту 1
* INCREMENT BY – шаг последовательности. По дефолту 1
* MAXVALUE – максимальное значение последовательности
* MINVALUE – минимальное значение последовательности
* NOMAXVALUE – устанавливает максимальное значение равным 1027 для возрастающей последовательности, или –1 для убывающей. NOMAXVALUE используется по умолчанию
* NOMINVALUE – определяет минимальное значение равное 1 для возрастающей последовательности и -1026 для убывающей. NOMINVALUE используется по умолчанию
* CYCLE – подразумевает, что последовательность начинает генерировать значения по кругу, при достижении максимального или минимального значения. При обращении к последовательности, когда она достигла максимального значения, следующее сгенерированное значение будет минимальным значением последовательности. В ситуации с убывающей последовательностью, при достижении минимального значения, следующее сгенерированное будет максимальным.
* CACHE – количество значений, сохраняемых в памяти. По умолчанию это 20. Минимальное количество кэшированных значений – 2
* NOCACHE – отключает кэширование. Это не позволит базе данных выделить некоторое количество значений заблаговременно, что позволит избежать пробелов в последовательности, но увеличит затраты системных ресурсов. Пробелы могут возникнуть при остановке базы данных, когда кэшированные значения теряются. Если опции CACHE и NOCACHE не указываются, то по умолчанию кэшируется 20 значений
* ORDER – обеспечивает генерацию значений в порядке запросов. Как правило, ORDER используется в среде Real Application Clusters.
* NOORDER – не дает таких гарантий. По умолчанию используется NOORDER

1. **Какие привилегии необходимы для создания и удаления последовательности?**

Привилегии create sequence достаточно для создания и удаления своей последовательности.

1. **Что такое кластер?**

А у вас бывало такое что вы типа работаете с двумя таблицами которые связаны по внешнему ключу и вас уже ну просто заебало что операция соединения таблиц занимает столько времени? Так вот, скажите спасибо ебать разрабам Оракла, ведь они придумали и на эту хуйню решение.

И называется оно кластер. Например, у вас есть две таблицы с одинаковым столбцом, на который вы их джоините. Было бы ебаловом хранить повторяющиеся значения на диске два раза; для этого создается кластер – единое пространство на диске, в котором хранятся данные из общих для двух таблиц столбцов.

**Кластер** – это физическое (на диске) совместное хранение двух и более таблиц с целью повышения производительности при операциях соединения. Есть и минусы – DML-операторы могут работать медленнее. Увеличение производительности происходит за счет уменьшения количества операций ввода-вывода с диска (которые, если что, являются одними из самых затратных по производительности)

1. **Что означает параметр hash?**

При необходимости можно также создавать хеш-кластеры. В таком случае строки таблицы извлекаются с помощью встроенной в Оракл хеш-функции. Для нахождения любой строки нужно всего лишь вычислеть хеш-значение по хеш-ключу. Думаю, всем очевидно, что такая операция производительнее, чем создавать обычные таблицы и вручную искать на диске нужную строчку.

Особенный прирост в производительности хеш-кластеры дают при операциях выборки с условием эквивалетности (i.e. те, в которых указывается WHERE column\_name = column\_value). Это связано с тем, что значение хеш-ключа непосредственно указывает на область диска,в которой база хранит строки.

1. **Какие привилегии необходимы для создания и удаления кластера?**

Create cluster.

1. **Что такое синоним?**

Синонимы – это просто псевдонимы (алиасы, ссылки) объектов базы данных. Могут указывать на любые объекты: таблицы, представления, процедуры, функции, последовательности, пакеты. Используются в основном для упрщения доступа пользователей к объектам, которые принадлежат другим юзерам, а также для безопасности.

Синонимы скрывают лежащие в них объекты. Они могут быть:

* публичными (public) – доступны всем пользователям базы данных
* частными (private) – принадлежат пользователю, их создавших, и являются частью схемы этого пользователя. Другим пользователям базы надо будет выдавать права доступа для использования частных синонимов.

1. **Чем отличается публичный синоним от частного синонима?**

Выше описано ёпта.

1. **Что такое материализованное представление?**

В чем идея: допустим, в компании есть база данных, и каждую неделю, квартал или год они хотят делать отчёт. У них есть представление, которое делает выборку по огромному объёму данных, и по итогу если им надо делать отчёт каждый день, то они сидят по 10 минут и ждут, пока он посчитается (если что, каждый раз при просмотре представления заново выполняется запрос, на основе которого представление было построено). Для решения подобной проблемы в Оракле была придумана такая вещь, как материализированные представления.

Смысл в том, что уже написанное материализированное представление будет обновляться каждый раз, когда обновляется соответствующая ему таблица, что ускоряет процесс выборки запросов (опцию автоматического обновления можно легко включить/выключить). Вообще, его можно представлять как специализированные представления, которые, в отличие от обычных представлений, имеющие физическое воплощение. То есть, они занимают место на диске и требуют хранения, подобно обычным таблицам, на них даже можно навесить индексы.

Также корнем материализированных представлений является механизм перезаписи запроса: Оракл автоматически перезапишет запрос таким образом, чтобы в нем использовались материализованные представления, существенно улучшая тем самым время ответа на запрос.

1. **Чем отличается материализованное представление от обычного представления?**

* Материализированное имеет место на диске (по нему даже можно создавать индексы)
* Обычные представления всегда выдают самые актуальные данные, а если редко обновлять материализированное представление, то данные в нём могут устареть (если не настроить автоматическое обновление)
* Материализированные имеют механизм перезаписи запроса (query rewrite), который автоматически использует готовое представление вместо новой операции выборки, если стоимость такого запроса будет ниже