1. **Расшифруйте аббревиатуру SGA. Перечислите основные пулы памяти SGA, поясните их назначение.**

Системная Глобальная область - SGA

Назначение SGA состоят в ускорении производительности запросов и обеспечении большого объема параллельной активности. Поскольку обработка в памяти намного быстрее дискового ввода-вывода, размер SGA – один из важнейших конфигурационных параметров при настройке базы данных на достижение оптимальной производительности. Когда вы запускаете экземпляр Oracle, он занимает определенный объем памяти из оперативной памяти операционной системы и этот объем определяется компонентом SGA в инициализационном файле. Когда экземпляр останавливается, память, использованная SGA, возвращается операционной системе.

SGA – наиболее важный компонент памяти в экземпляре Oracle.

* Буферный кэш базы данных. Хранит копии блоков данных, прочитанных из файлов данных.
* Разделенный пул. Содержит библиотечный кэш для хранения, разобранного SQL и PL/SQL кода, готового к использованию всеми пользователями. Он также содержит кэш словаря данных, который хранит всю информацию словаря.
* Буфер журнала повторного выполнения. Содержит информацию, необходимую для восстановления изменений, проведенных в базе данных операциями DML (языка манипулирования данными). Эта информация затем записывается в журналы повторного выполнения писателем журналов.
* Пул Java. Представляет пространство «кучи» для создания объектов Java.
* Большой пул. Хранит крупные выделения памяти, такие как резервные буферы RMAN.
* Пул потоков. Поддерживает средство Oracle Streams (средство для репликации данных между базами данных).

1. **Поясните параметры SGA\_MAX\_SIZE и SGA\_TARGET.**

**SGA\_MAX\_SIZE** задает максимальный размер SGA для времени жизни экземпляра.

**SGA\_TARGET** – указывает текущий (возможный) размер памяти

1. **Поясните назначение буферного кэша инстанса. Поясните назначение пулов КЕЕP, DEFAULT и RECYCLE буферного кэша.**

Буферный кэш базы данных состоит из буферов памяти, которые Oracle использует для хранения данных, прочитанных серверным процессом из файлов данных на диске в ответ на запросы пользователей. Доступ к буферному кэшу, конечно же, осуществляется намного быстрее, чем чтение данных из дискового хранилища. Когда пользователь модифицирует данные, эти изменения проводятся также в буферном кэше базы данных. Поэтому буферный кэш содержит как оригинальные блоки, прочитанные с диска, так и измененные блоки, которые подлежат записи на диск.

|  |  |  |
| --- | --- | --- |
| **Буферный пул** | **Инициализационный параметр** | **Описание** |
| Постоянный буферный пул(keep buffer pool) | *DB\_KEEP\_CACHE\_SIZE* | Постоянно хранит блоки данных в памяти. У вас могут быть маленькие таблицы, к которым выполняются частые обращения, и для предотвращения их удаления из буферного кэша им можно назначить постоянный буферный пул при создании таблицы. |
| Повторно используемый буферный пул(recycle buffer pool) | *DB\_RECYCLE\_CACHE\_SIZE* | Удаляет данные из кэша немедленно после использования. Этот буферный пул следует применять осторожно, если вы вообще решите использовать его. Повторно используемый буферный пул удаляет объект из кэша сразу по завершении транзакции. Очевидно, что его следует применять только для крупных таблиц, обращение к которым осуществляется нечасто, и которые не нужно хранить к кэше неопределенно долго. |
| Буферный пул по умолчанию(default buffer pool) | *DB\_CACHE\_SIZE* | Содержит все данные и объекты, которые не назначены в постоянный и повторно используемый буферные пулы. |

1. **Поясните принцип вытеснения блоков буферного кэша (LRU).**

Буферы в кэше организованы в два списка: список записи и список наименее недавно использованных (LRU). Список **записи** содержит грязные буферы*,* которые содержат данные, которые были изменены, но еще не были записаны на диск. Список **LRU** содержит свободные буферы, закрепленные буферы и грязные буферы, которые еще не были перемещены в список записи. **Свободные буферы** не содержат никаких полезных данных и доступны для использования. В настоящее время осуществляется доступ к закрепленным буферам.

В первый раз, когда пользовательскому процессу Oracle требуется определенный фрагмент данных, он выполняет поиск данных в буферном кэше базы данных. Если процесс находит данные уже в кэше (**попадание** в кэш), он может считывать данные непосредственно из памяти. Если процесс не может найти данные в кэше (**промах** кэша), он должен скопировать блок данных из файла данных на диске в буфер в кэше, прежде чем обращаться к данным. Доступ к данным через попадание в кэш происходит быстрее, чем доступ к данным через промах кэша.

**Least recently used** (LRU): в первую очередь, вытесняется неиспользованный дольше всех. Этот алгоритм требует отслеживания того, что и когда использовалось, что может оказаться довольно накладно, особенно если нужно проводить дополнительную проверку, чтобы в этом убедиться. Общая реализация этого метода требует сохранения «бита возраста» для строк кэша и за счет этого происходит отслеживание наименее использованных строк (то есть за счет сравнения таких битов).

1. **Поясните принцип вытеснения блоков таблицы, созданной оператором CREATE TABLE … CACHE.**

CACHE – помещение таблицы в конец LRU-списка (для малых таблиц) обычно в default pool

Если вы последовательно выполните один и тот же запрос два или более раз, вы обратите внимание, что второй (и последующие) запросы выполняются быстрее, чем первый. Дело в том, что начальный запрос перемещает данные таблицы в буфера данных в памяти, где они остаются, пока не будут вытолкнуты другими данными из других таблиц, используемых для ответа на другие запросы. Чтобы сохранить конкретную таблицу в памяти, вы должны закрепить ее в кэше. Это стоит делать только с малыми таблицами, которые находятся в постоянном использовании.

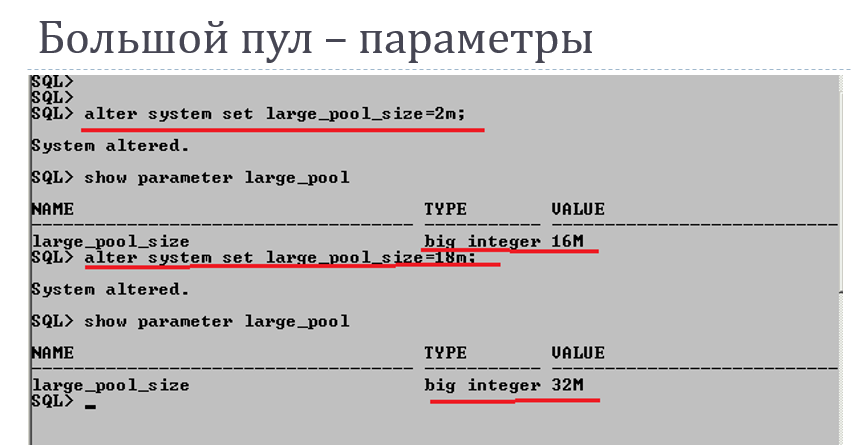
Если вы обнаружили, что сервер выталкивает основные таблицы из памяти, вы можете закрепить их в памяти, используя параметр CACHE оператора CREATE/ALTER TABLE:

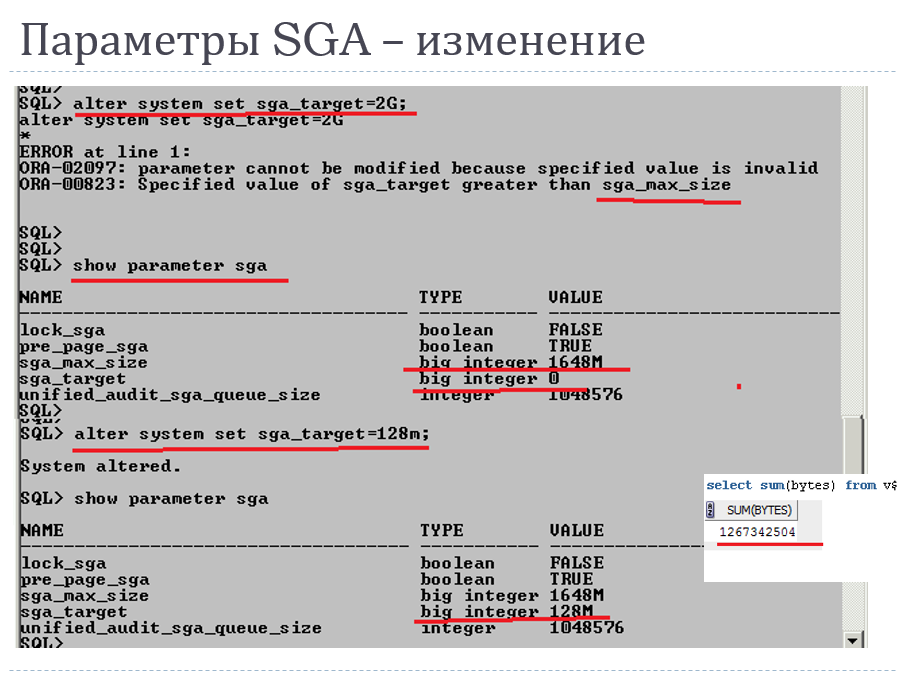
ALTER TABLE [TABLENAME] CACHE;

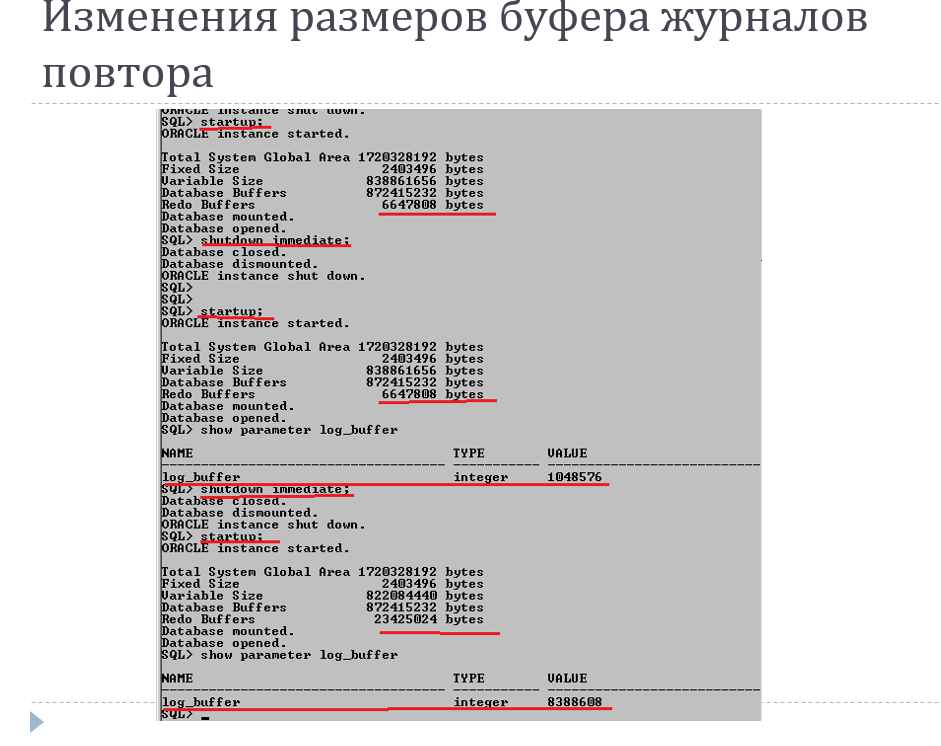
Table altered.

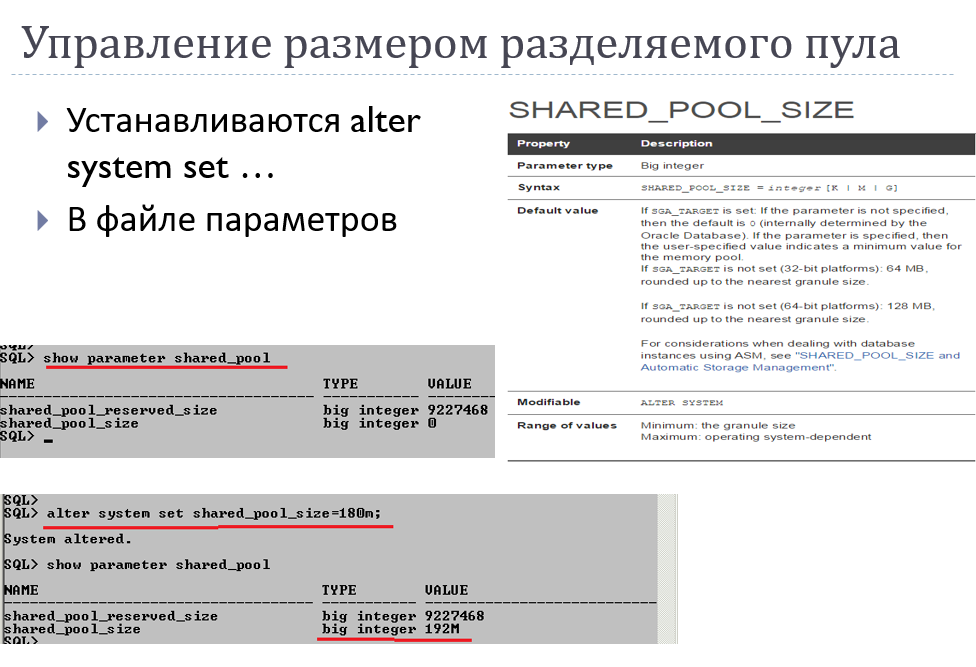
Этот параметр гарантирует, что данные из таблицы после полного ее сканирования находятся в списке самых недавно использованных (most recently used - MRU) данных, а не в списке самых давно использованных (least recently used - LRU) данных, в результате чего они будут сохранены в памяти для последующего использования.

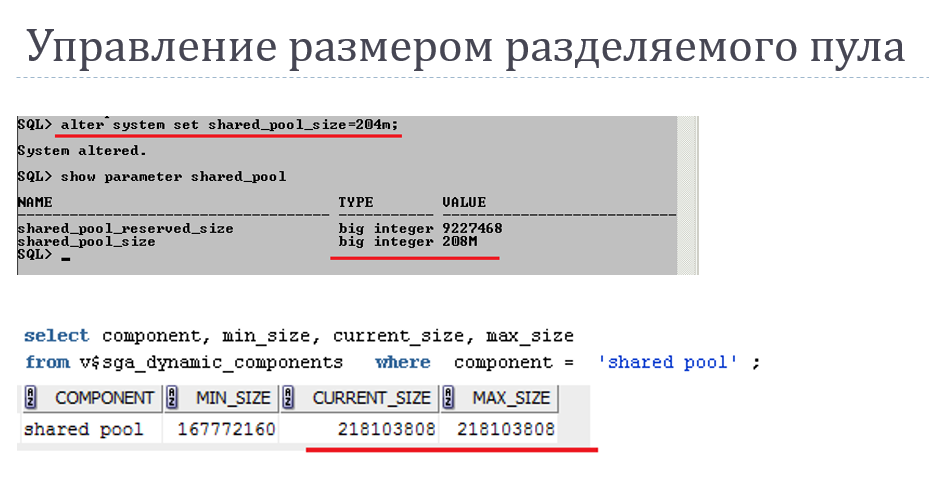
1. **Как изменить размеры пулов?**













1. **Какие пулы допускают изменение размеров?**

Oracle может автоматически управлять следующими пятью компонентами SGA (соответствующие инициализационные параметры Oracle указаны в скобках):

* буферный кэш базы данных (*DB\_CACHE\_SIZE*);
* разделяемый пул (*SHARED\_POOL\_SIZE*);
* большой пул (*LARGE\_POOL\_SIZE*);
* пул Java (*JAVA\_POOL\_SIZE*);
* пул потоков (*STREAMS\_POOL\_SIZE*).

Как видите, Oracle автоматически настраивает пять компонентов SGA, которые мы называем параметрами SGA с автоматически устанавливаемым размером. Вы должны по-прежнему самостоятельно управлять остальными компонентами SGA, даже при автоматическом управлении памятью.