# Лабораторная работа 1. Архитектура сервера Oracle

1. Структура памяти:

Select \* from v$SGA; - отображает сводную информацию о глобальной области системы (SGA).

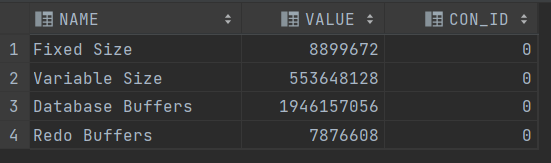


Рисунок 1 – сводная информация.

select \* from v$sgainfo; - отображает информацию о размере SGA.

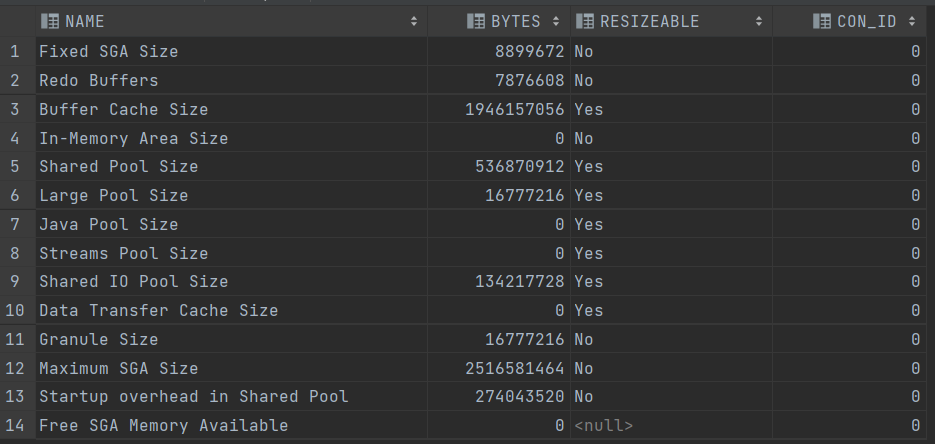


Рисунок 2 – размеры компонентов SGA, возможность изменения, ID контейнера

select component,current\_size,min\_size,max\_size from v$sga\_dynamic\_components; - отображает информацию о динамических компонентах SGA. В этом представлении обобщается информация, основанная на всех выполненных операциях изменения размера SGA с момента запуска экземпляра.

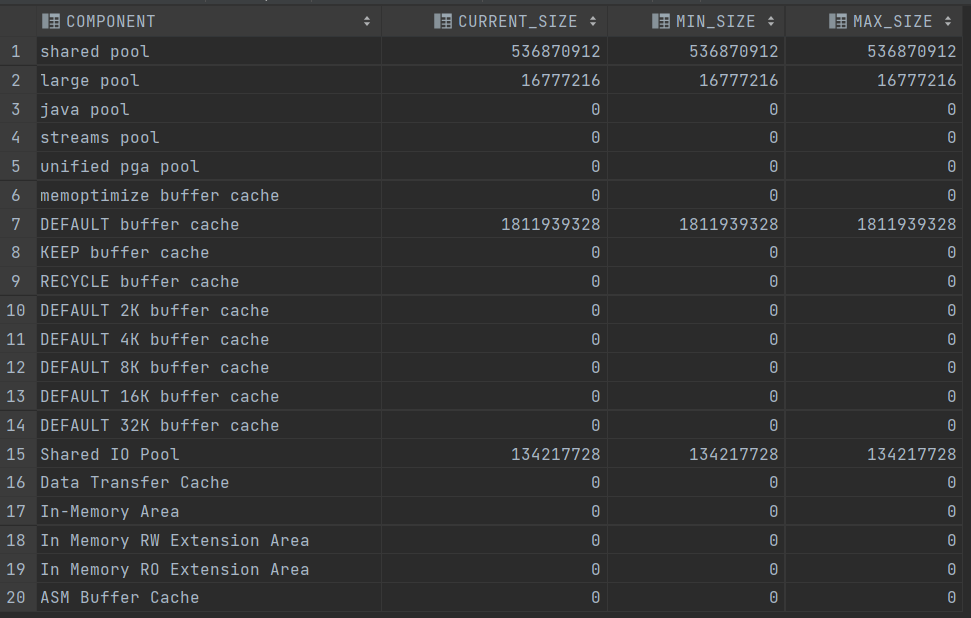


Рисунок 3 – информация о динамических объектах: минимальный, максимальный и текущий размер.

1. Файлы базы данных:

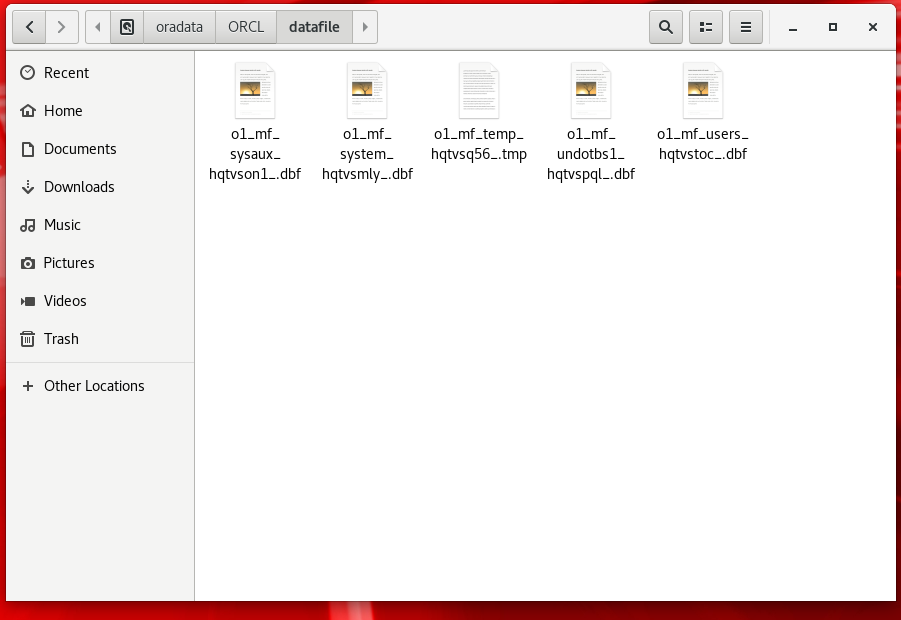
select tablespace\_name, file\_name from dba\_data\_files;

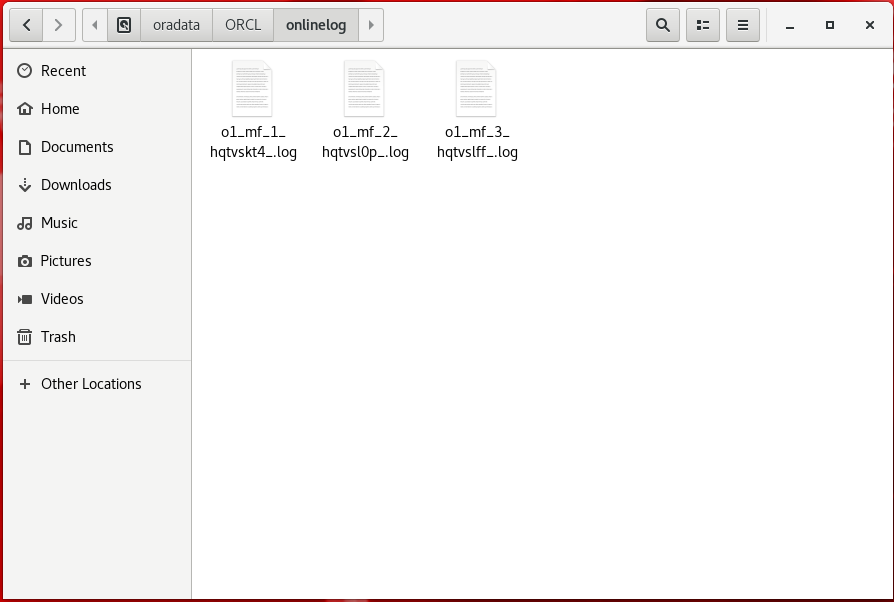
select tablespace\_name, file\_name from dba\_temp\_files;

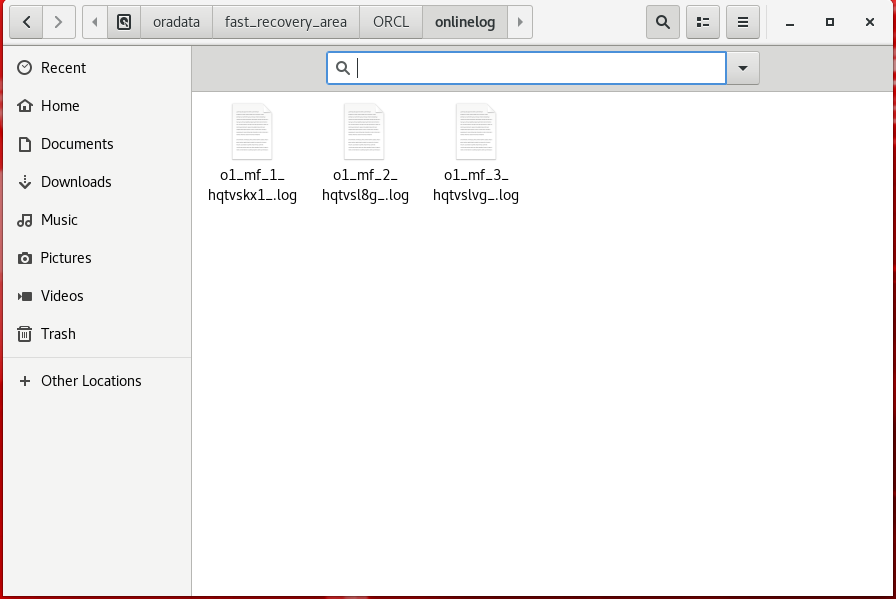
select group#,member from v$logfile;

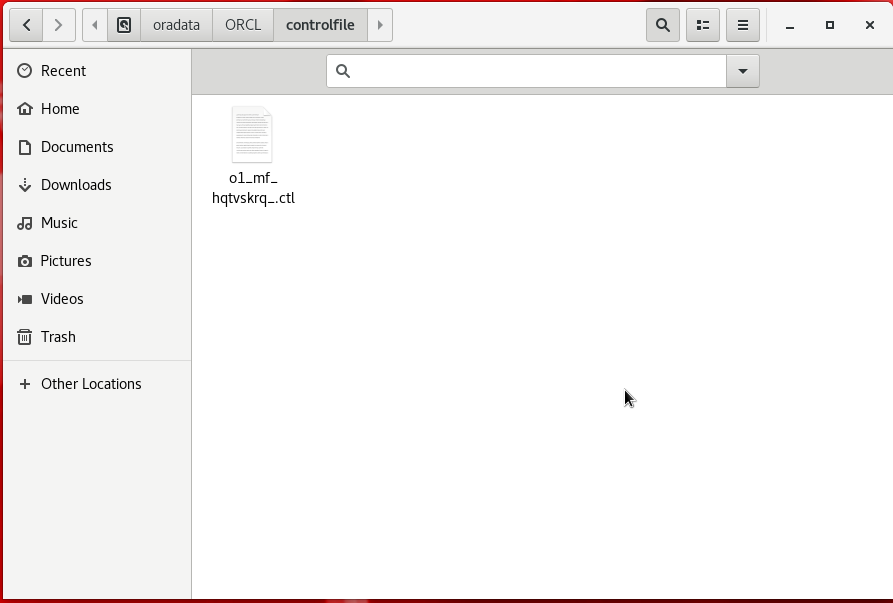
select name from v$controlfile;

Найти файлы на диске. Рассмотреть структуры системных таблиц и представлений с т.з. получения информации.

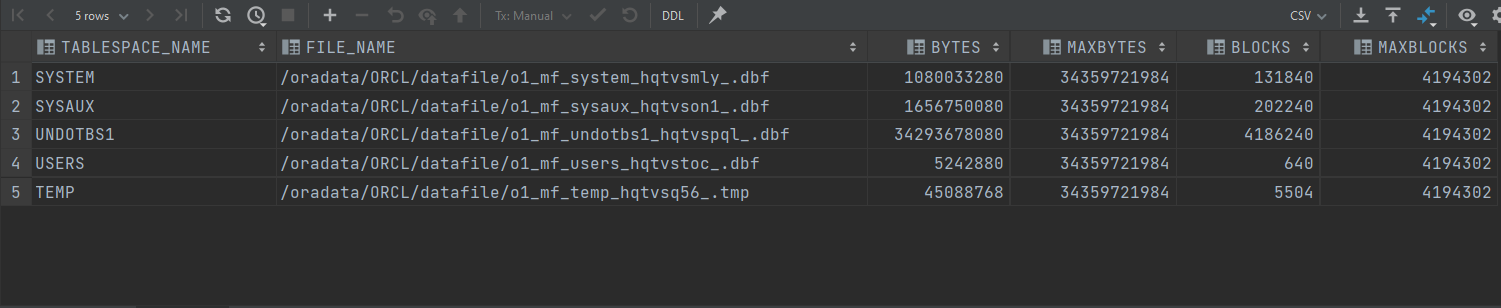






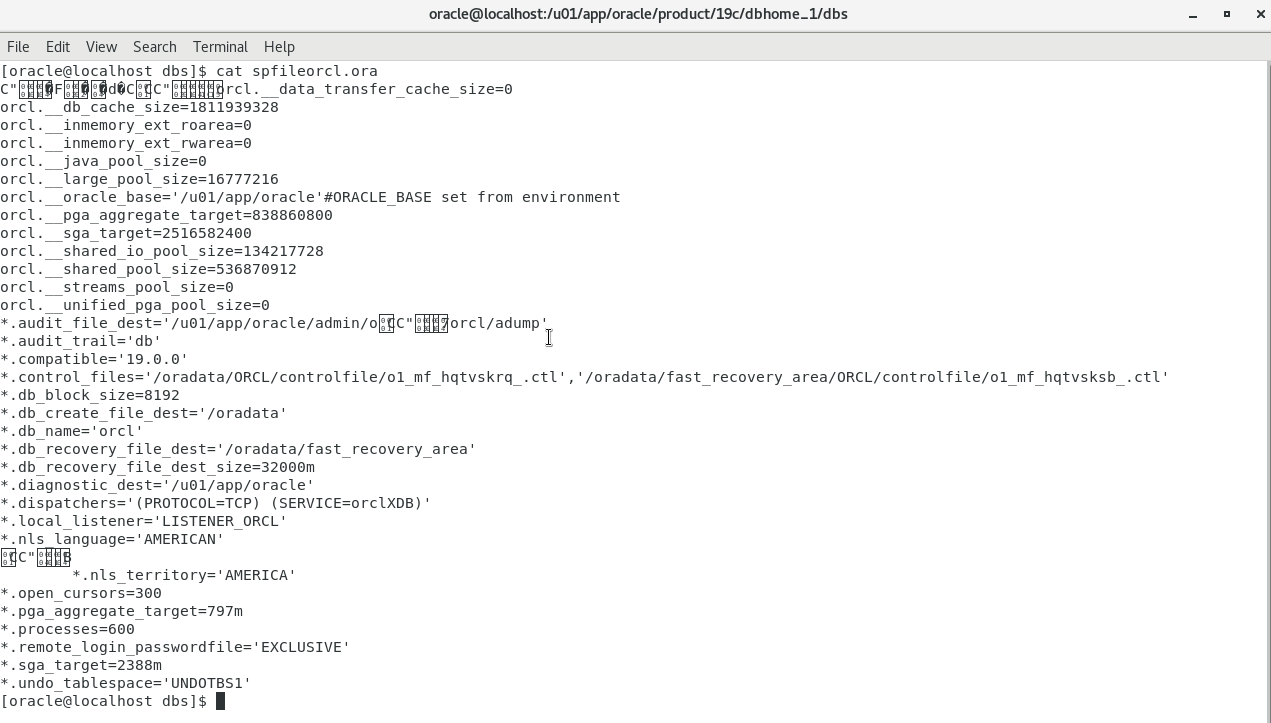


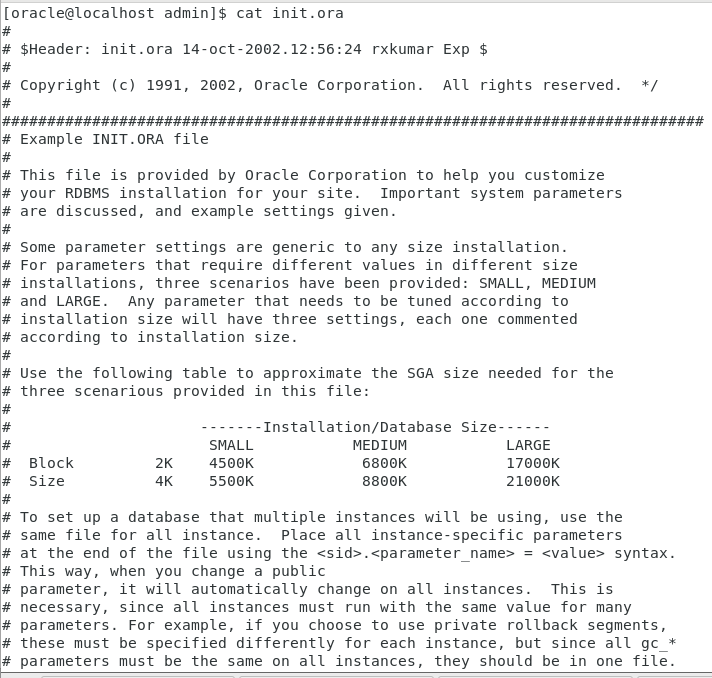
* Какое кол-во блоков выделено для каждого табличного пространства?
* Фактически занято?

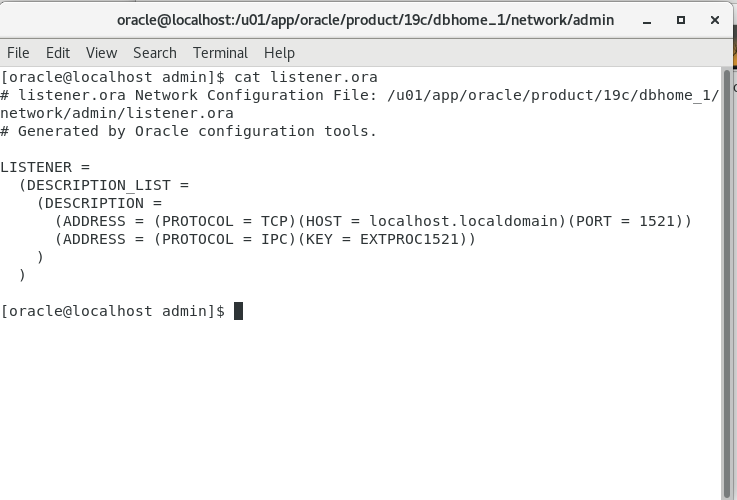


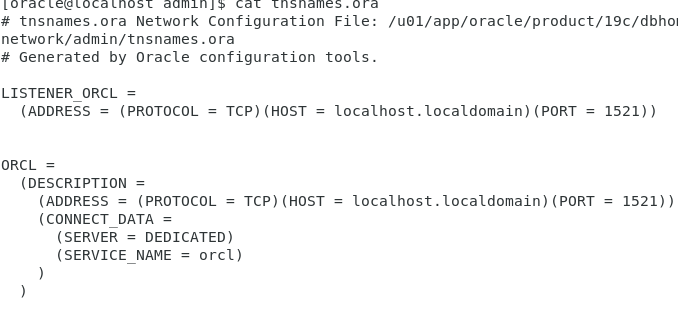
1. Содержимое основных конфигурационных файлов.

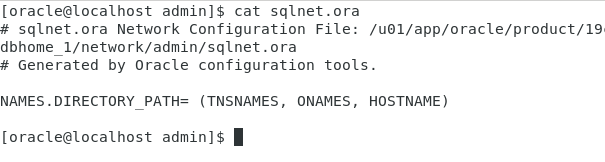
SPFileXE, pfile, PWDXE, initXE, listener, tnsnames, sqlnet





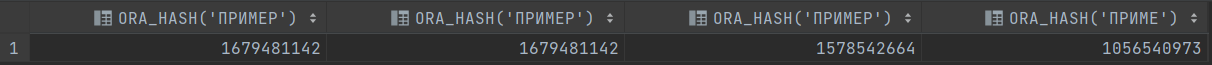






1. Выполнить анализ работы HASH-функции Oracle.

SELECT ora\_hash('ПРИМЕР'), ora\_hash('ПРИМЕр'),ora\_hash('ПРИМЕР '),ora\_hash('ПРИМЕ') FROM dual; - ora\_hash() – функция рассчитывающая хэш. Хэш не меняется у одинаковых объектов.



1. Выполнить анализ содержимого Library\_Cashe представление v$SQL.

SELECT \* /\* myQuery\*/ FROM EMP WHERE ename= ‘SCOTT’; -- 1

SELECT \* /\* myQuery\*/ FROM EMP WHERE ename= ‘JONES’; -- 2

SELECT \* /\* myQuery\*/ FROM EMP WHERE ename= :nn; -- 3

Указав:**BLAKE**

SELECT \* /\* myQuery\*/ FROM EMP WHERE ename= :nn; -- 4

Указав:**KING**

Выполнить анализ содержимого Library\_Cashe:

Select \* from v$SQL where sql\_text like '%myQuery%';

Запросы 1 и 2 разные, а 3 и 4 одинаковые. Параметр можно использовать для оптимизации выполнения запросов.

1. Получить перечень процессов, выполняемых в эту сессию на сервере.

select \* from v$process;

1. Рассмотреть запросы для таблиц и представлений, приведенных в приложении.

Вопросы к лекции 2 «Архитектура ORACLE»

1. Какие фазы обработки SQL команды?

Parse, bind, execute, fetch.

1. Как происходит разбор SQL команды?

**Мягкий разбор:**Представленное SQL выражение имеется в sharedpool. Оно может быть помещено туда в процессе его предыдущего выполнения. В этом случае используется уже разобранная версия выражения, что предотвращает его повторный разбор.

**Жесткий разбор:**Представленное SQL выражение выполняется впервые и не имеет доступной копии разобранного кода в sharedpool. Операция жесткого разбора наиболее ресурсоёмкая  поскольку в процессе разбора выполняются в полном обьеме все операции parse.

1. Построить запрос, который выведет данные в следующем виде

|  |  |
| --- | --- |
| **SQL\_TEXT** | **NAME** |
| **select \* /\*Test\*/ from emp where deptno=:Dprt** | **:DPRT,** |

SELECT  
 E.sql\_text,  
 S.NAME  
FROM  
 V$SQL E  
 full join V$SQL\_BIND\_CAPTURE S  
 on E.SQL\_ID = S.SQL\_ID  
WHERE E.sql\_text like '/\*Test\*/';

1. Какие процессы задействованы при выполнении DML-команды?

DML выражения выполняются в две фазы: Parse в процессе которой разбирается выражение и фаза Execute, в которую для DML включены операции захвата данных и сортировки.

Процесс выполнения DML выражения:

1. Если данные и блоки отката для изменяемых в процессе выполнения DML данных еще не помещены в буферный кэш, серверный процесс считывает их с диска в буферный кэш. Серверный процесс при этом блокирует строки, которые будут модифицированы.

2. Серверный процесс затем применяет изменения данных, запрашиваемые в DML к данным, прочитанным в буферный кэш, и записывает данные в буфер отката. Изменения записываются в redologbufferперед тем как данные будут изменены в буферном кэше. Эта технология применения изменений называется "упреждающее логирование" (write-aheadlogging).

3. Буферы отката содержат значения данных перед модификацией. Буферы отката используются для хранения снимка данных до изменения и таким образом изменения,внесенные DML операциями, могут быть отменены в случае необходимости. В буферы данных записываются изменения данных.

4. Пользователь получает ответ от операции DML (сколько строк изменено в процессе выполнения операции).

1. В каком порядке выполняется фиксация изменений данных в таблицах базы по команде COMMIT?

• Серверный процесс помещает запись о commit c SCN в redologbuffer.

• Фоновый процесс LGWR записывает исходные данные в redolog файлы. Это гарантирует, что в случае отказа экземпляра, данные об изменениях произведенных транзакцией не будут потеряны.

• Серверный процесс отправляет сообщение об окончании транзакции пользовательскому процессу.