Вопросы к лекции 2 «Архитектура ORACLE»

1. Какие фазы обработки SQL команды?
2. Как происходит разбор SQL команды?
3. Построить запрос, который выведет данные в следующем виде

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

если исходный запрос был:

***select \* /\*Test\*/ from emp where deptno=:Dprt;***

***Внимание***: переменных связи может быть несколько, для каждой переменной своя строка в таблице результатов.

1. Какие процессы задействованы при выполнении DML-команды?
2. В каком порядке выполняется фиксация изменений данных в таблицах базы по команде COMMIT?

ЗАДАНИЕ 1

Разбор происходит следующим образом:

1. Проверяет выражение на правильность семантики и синтаксиса
2. Определят есть ли права доступа на транзакцию у пользователя
3. Ищет совпадения выполняемого кода с уже разобранным кодом, хранящимся в библиотечном кеше
4. Выделяет частную SQL область для обработки выражения

ЗАДАНИЕ 2

Существует два вида разбора:

1. **Мягкий разбор:**Представленное SQL выражение имеется в shared pool. Оно может быть помещено туда в процессе его предыдущего выполнения. В этом случае используется уже разобранная версия выражения, что предотвращает его повторный разбор.
2. **Жесткий разбор:**Представленное SQL выражение выполняется впервые и не имеет доступной копии разобранного кода в shared pool. Операция жесткого разбора наиболее ресурсоёмкая  поскольку в процессе разбора выполняются в полном обьеме все операции parse.

ЗАДАНИЕ 3 – нужно будет переделать запрос

Select \* from v$SQL where sql\_text like '%select \* /\*Test\*/ from emp where deptno=:Dprt%'

ЗАДАНИЕ 4

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 redo log buffer **перед тем как** данные будут изменены в буферном кэше. Эта технология применения изменений называется "упреждающее логирование" (write-ahead logging)
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)

ЗАДАНИЕ 5

Когда запускается COMMIT, выполняются следующие шаги:

* Серверный процесс помещает запись о commit c SCN в redo log buffer.
* Фоновый процесс LGWR записывает исходные данные в redo log 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.

DBWR в итоге, записывает актуальные данные из грязных блоков на диск используя для этого свой внутренний временной механизм и настойки выполнения алгоритма checkpoint. Команда обратная commit – это ROLLBACK