

Без использования Enterprise Manager для любой произвольной команды SELECT определить рекомендации по улучшению ее производительности.  
Например, для команды 'SELECT COUNT(\*) FROM SH.SALES' результат должен быть:

Команда
CREATE MATERIALIZED VIEW LOG
CREATE MATERIALIZED VIEW
GATHER TABLE STATISTICS

Для выполнения данной задачи буду использовать Sql Tunning Set:

SQL Tuning Set (STS) - это структура данных в базе данных Oracle, которая используется для сбора, анализа и оптимизации SQL-запросов. Она представляет собой набор SQL-запросов, которые могут быть выполнены на базе данных.

STS содержит следующие компоненты:

1. SQL statements (SQL-запросы) - Это основные SQL-запросы, которые собираются и хранятся в STS. Это могут быть как выполняющиеся запросы, так и запросы, которые еще не были выполнены, но могут быть выполнены в будущем.
2. Bind values (значения переменных) - Если SQL-запрос содержит переменные, такие как WHERE-условия с плейсхолдерами, то значения этих переменных также сохраняются в STS. Это позволяет базе данных анализировать выполнение запросов с разными значениями переменных и оптимизировать их выполнение.
3. Execution statistics (статистика выполнения) - Для каждого SQL-запроса в STS сохраняются данные о его выполнении, такие как количество выполнений, время выполнения, количество обращений к диску и другие статистические показатели. Это помогает анализировать производительность запросов и оптимизировать их выполнение.

STS может быть использован для оптимизации SQL-запросов путем выполнения следующих действий:

1. Анализ производительности запросов - STS позволяет анализировать производительность SQL-запросов и определить узкие места или проблемные запросы, которые требуют оптимизации.
2. Оптимизация запросов - STS предоставляет возможность изменять и оптимизировать SQL-запросы, чтобы улучшить их производительность. Можно редактировать SQL-запросы напрямую в STS или использовать инструменты для автоматической оптимизации.
3. Сравнение производительности разных вариантов запросов - STS позволяет сравнивать производительность разных вариантов SQL-запросов с разными значениями переменных. Это позволяет выбрать наилучший вариант запроса для выполнения.

Использование SQL Tuning Set может значительно повысить производительность SQL-запросов в базе данных Oracle и помочь оптимизировать их выполнение

1. Зададим несколько запросов для тестирования:

```
SQL> conn sys/oracle@student as sysdba
Connected.
SQL> Select count(*) from SH.sales;

COUNT(*)
-----
918843

SQL> Select count(*) from hr.employees where department_id=100;

COUNT(*)
-----
6

SQL> insert into hr.dept values(800,'800',120,10);

1 row created.
```

2. Команда EXECUTE DBMS\_SQLTUNE.CREATE\_SQLSET(SQLSET\_NAME => 'dop45', SQLSET\_OWNER => 'SYS'); выполняет создание SQL-набора (SQL set) с именем 'dop45' в схеме 'SYS' с использованием пакета DBMS\_SQLTUNE.

SQL-набор (SQL set) представляет собой объект, который используется для группировки и управления набором SQL-запросов, которые требуют анализа и оптимизации в Oracle.

```
1 row created.

SQL> execute dbms_sqltune.create_sqlset(sqlset_name => 'dop45', sqlset_owner =>
'SYS');

PL/SQL procedure successfully completed.

SQL> begin
2   DBMS_SCHEDULER.CREATE_JOB (
3     job_name => 'CREATE_JOB_SELECT4_2',
4     job_type => 'PLSQL_BLOCK',
5     job_action =>
6       'DECLARE
7         sqlset_cur dbms_sqltune.sqlset_cursor;
8         str VARCHAR2(50);
9         v_value NUMBER;
10
11        BEGIN
12          str := q''[upper(SQL_TEXT) LIKE ''SELECT%'']'';
13          OPEN sqlset_cur FOR
14            SELECT VALUE(P)
15              FROM TABLE(dbms_sqltune.select_cursor_cache(attribute_1
16                ist=>'TYPICAL', basic_filter=>str)) P;
17          dbms_sqltune.load_sqlset(
18            sqlset_name=>'dop45',
19            populate_cursor=>sqlset_cur,
20            load_option=>'MERGE',
21            update_option=>'ACCUMULATE',
22            sqlset_owner=>'SYS'
23          );
24        END;
25      start_date      => SYSTIMESTAMP,
26      repeat_interval => 'FREQ=MINUTELY; INTERVAL=1',
27      end_date        => SYSTIMESTAMP + INTERVAL '1' DAY,
28      enabled         => TRUE);
29 end;
30 /

PL/SQL procedure successfully completed.

SQL> _
```

И создадим планировщик, который будет загружать в набор новые запросы:

1. Объявление переменных sqlset\_cur, str и v\_value.
2. Инициализация переменной str с использованием синтаксиса q"[...]" для определения строки как текста запроса. В данном случае, строка str содержит условие фильтрации для basic\_filter, которое ограничивает результаты выборки запросами, начинающимися с ключевого слова "SELECT".
3. Открытие курсора sqlset\_cur с помощью функции DBMS\_SQLTUNE.SELECT\_CURSOR\_CACHE(). Этот курсор содержит результат выборки, соответствующей условию фильтрации str, и используется для загрузки данных в SQL-набор.
4. Загрузка данных в SQL-набор 'dop45' с помощью вызова процедуры DBMS\_SQLTUNE.LOAD\_SQLSET(). Здесь используются следующие параметры:
  - sqlset\_name - имя SQL-набора, в данном случае 'dop45'.
  - populate\_cursor - курсор, содержащий результаты выборки, который будет загружен в SQL-набор. В данном случае, это sqlset\_cur.

- load\_option - опция загрузки, в данном случае 'MERGE', что означает объединение данных. Опция ACCUMULATE указывает на то, что новые данные, передаваемые в SQL-набор, будут добавляться (накапливаться) к уже существующим данным в SQL-наборе. Это означает, что существующие данные в SQL-наборе не будут заменены, а будут обновляться путем добавления новых данных.

3. Далее создаем задачу для advisor:

-- Создание задачи SQL Access Advisor

```
dbms_advisor.create_task(DBMS_ADVISOR.SQLACCESS_ADVISOR, task_id, taskname, task_desc,  
task_or_template);
```

-- Сброс задачи

```
dbms_advisor.reset_task(taskname);
```

-- Проверка наличия связи между задачей и набором SQL

```
SELECT count(*)  
INTO num_found  
FROM user_advisor_sqla_wk_map  
WHERE task_name = taskname  
AND workload_name = sts_name;
```

-- Удаление ссылки на набор SQL при наличии связи

```
IF num_found > 0 THEN  
    dbms_advisor.delete_sts_ref(taskname, sts_owner, sts_name);  
END IF;
```

-- Добавление ссылки на набор SQL

```
dbms_advisor.add_sts_ref(taskname, sts_owner, sts_name);
```

1. dbms\_advisor.set\_task\_parameter(taskname, 'INVALID\_SQLSTRING\_LIST', '@!');

- Устанавливает параметр INVALID\_SQLSTRING\_LIST для указанной задачи.

- Значение параметра INVALID\_SQLSTRING\_LIST установлено как '@!', что указывает на список невалидных SQL-выражений.

- В этом случае, невалидные SQL-выражения будут иметь значение '@!'.

2. dbms\_advisor.set\_task\_parameter(taskname, 'ANALYSIS\_SCOPE', 'ALL');

- Устанавливает параметр ANALYSIS\_SCOPE для указанной задачи.

- Значение параметра ANALYSIS\_SCOPE установлено как 'ALL', что указывает, что область анализа включает все доступные объекты.

3. dbms\_advisor.set\_task\_parameter(taskname, 'RANKING\_MEASURE', 'PRIORITY, OPTIMIZER\_COST');

- Устанавливает параметр RANKING\_MEASURE для указанной задачи.

- Значение параметра RANKING\_MEASURE установлено как 'PRIORITY, OPTIMIZER\_COST', что указывает на меры ранжирования по приоритету и стоимости оптимизатора.

4. dbms\_advisor.set\_task\_parameter(taskname, 'TIME\_LIMIT', 10000);

- Устанавливает параметр TIME\_LIMIT для указанной задачи.

- Значение параметра TIME\_LIMIT установлено как 10000, что указывает максимальное время выполнения задачи в миллисекундах.

5. dbms\_advisor.set\_task\_parameter(taskname, 'MODE', 'LIMITED');

- Устанавливает параметр MODE для указанной задачи.

- Значение параметра MODE установлено как 'LIMITED', что указывает на режим выполнения задачи в ограниченном режиме.

6. `dbms_advisor.set_task_parameter(taskname, 'STORAGE_CHANGE', DBMS_ADVISOR.ADVISOR_UNLIMITED);`

- Устанавливает параметр `STORAGE_CHANGE` для указанной задачи.
- Значение параметра `STORAGE_CHANGE` установлено на `DBMS_ADVISOR.ADVISOR_UNLIMITED`, указывая, что нет ограничений на изменение хранилища.

7. `dbms_advisor.set_task_parameter(taskname, 'DML_VOLATILITY', 'TRUE');`

- Устанавливает параметр `DML_VOLATILITY` для указанной задачи.
- Значение параметра `DML_VOLATILITY` установлено как `'TRUE'`, указывая, что волатильность DML должна быть учтена при анализе и рекомендациях.

8. `dbms_advisor.set_task_parameter(taskname, 'WORKLOAD_SCOPE', 'PARTIAL');`

- Устанавливает параметр `WORKLOAD_SCOPE` для указанной задачи.
- Значение параметра `WORKLOAD_SCOPE` установлено как `'PARTIAL'`, указывая, что область нагрузки будет ограничена.

9. `dbms_advisor.set_task_parameter(taskname, 'CREATION_COST', 'TRUE');`

- Устанавливает параметр `CREATION_COST` для указанной задачи.
- Значение параметра `CREATION_COST` установлено как `'TRUE'`, указывая, что затраты на создание будут учтены при рекомендациях.

10. `dbms_advisor.set_task_parameter(taskname, 'JOURNALING', '4');`

- Устанавливает параметр `JOURNALING` для указанной задачи.
- Значение параметра `JOURNALING` установлено как `'4'`, указывая уровень детализации журналирования задачи.

11. `dbms_advisor.set_task_parameter(taskname, 'DAYS_TO_EXPIRE', '30');`

- Устанавливает параметр `DAYS_TO_EXPIRE` для указанной задачи.
- Значение параметра `DAYS_TO_EXPIRE` установлено как `'30'`, указывая срок действия задачи в течение 30 дней.

```

SQL> DECLARE
2  taskname varchar2(30):='TaskForSelectAdvisor';
3  task_desc varchar2(256):='SQL Access Advisor';
4  task_or_template varchar2(30):='SQLACCESS_EMTASK';
5  task_id number := 0;
6  num_found number;
7  sts_name varchar2(256) := 'dop45';
8  sts_owner varchar2(30) := 'SYS';
9  BEGIN
10 dbms_advisor.create_task(DBMS_ADVISOR.SQLACCESS_ADVISOR, task_id,taskname,
task_desc, task_or_template);
11 dbms_advisor.reset_task(taskname);
12 Select count(*) into num_found from user_advisor_sqla_wk_map where task_name = taskname and workload_name = sts_name;
13 IF num_found > 0 THEN
14 dbms_advisor.delete_sts_ref(taskname, sts_owner, sts_name);
15 END if;
16 dbms_advisor.add_sts_ref (taskname,sts_owner, sts_name);
17 dbms_advisor.set_task_parameter (taskname, 'INVALID_SQLSTRING_LIST' , '@!'
');
18 dbms_advisor.set_task_parameter (taskname, 'ANALYSIS_SCOPE' , 'ALL');
19 dbms_advisor.set_task_parameter (taskname, 'RANKING_MEASURE' , 'PRIORITY,
OPTIMIZE cost');
20 dbms_advisor.set_task_parameter (taskname, 'TIME_LIMIT' ,10000);
21 dbms_advisor.set_task_parameter (taskname, 'MODE' , 'LIMITED');
22 dbms_advisor.set_task_parameter (taskname, 'STORAGE_CHANGE' , DBMS_ADVISOR
.ADVISOR_UNLIMITED) ;
23 dbms_advisor.set_task_parameter (taskname, 'DML_VOLATILITY' , 'TRUE');
24 dbms_advisor.set_task_parameter (taskname, 'WORKLOAD_SCOPE' , 'PARTIAL');
25 dbms_advisor.set_task_parameter (taskname, 'CREATION_COST' , 'TRUE');
26 dbms_advisor.set_task_parameter (taskname, 'JOURNALING' , '4');
27 dbms_advisor.set_task_parameter (taskname, 'DAYS_TO_EXPIRE' , 30);
28 end;
29
30 /
PL/SQL procedure successfully completed.
SQL>

```

```

SQL> execute dbms_advisor.execute_task('TaskForSelectAdvisor');
PL/SQL procedure successfully completed.

```

Вызовем задачу:

4. Проверим результаты обработки:

```

SQL>
SQL> select command, sql_text
2  from dba_advisor_actions join DBA_ADVISOR_SQLA_WK_STMTS using (rec_id,task_
name)
3  where task_name='TaskForSelectAdvisor';

COMMAND
-----
SQL_TEXT
-----

CREATE MATERIALIZED VIEW LOG
Select count(*) from SH.sales

CREATE MATERIALIZED VIEW
Select count(*) from SH.sales

GATHER TABLE STATISTICS
Select count(*) from SH.sales

COMMAND
-----
SQL_TEXT
-----

RETAIN INDEX
Select count(*) from hr.employees where department_id=100

SQL>

```

Только для select команд выданы рекомендации.

Проанализированы же были больше команд:

```
SQL> select task_name, sql_text
       2 from DBA_ADVISOR_SQLA_WK_STMTS;

TASK_NAME
-----
SQL_TEXT
-----

TaskForSelectAdvisor
select 0 CAPTURE_COUNT, 0 APPLY_COUNT, 0 PROP_COUNT, 0 CAPTURE_ERROR_COUNT, 0 AP

TaskForSelectAdvisor
Select count(*) from SH.sales

TaskForSelectAdvisor
SELECT :B3 , :B2 , :B1 FROM DUAL

TASK_NAME
-----
SQL_TEXT
-----

TaskForSelectAdvisor
SELECT 1 COL1, A.KEY_COLUMN_VALUE,A.METRIC_COLUMN,A.METRIC_VALUE FROM TABLE(CAST

TaskForSelectAdvisor
SELECT COLUMN_NAME, DATA_TYPE, POSITION, DATA_DEFAULT FROM THE(SELECT CAST(:B1 A

TaskForSelectAdvisor
select 1 from dual where :INIT_PARAM_VALUE = 'FALSE'

TASK_NAME
-----
SQL_TEXT
-----

TaskForSelectAdvisor
SELECT SYS_EXTRACT_UTC(SYSTIMESTAMP) FROM DUAL

TaskForSelectAdvisor
SELECT SYS_GUID() FROM SYS.DUAL

TaskForSelectAdvisor
select 1 from dual where :CONTROL_FILE_COUNT = 1

TASK_NAME
-----
SQL_TEXT
-----

TaskForSelectAdvisor
select 1 from dual where UPPER(:pwd_status) <> 'ON'

TaskForSelectAdvisor
select 1 from dual where :package = 'DBMS_LOB'
```

#### Список литературы:

1. <https://oracle-patches.com/db/sql/%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82-sql-tuning-advisor-%D0%B4%D0%BB%D1%8F-%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8-%D0%BC%D0%B5%D0%B4%D0%BB%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE-%D0%BA%D0%BE%D0%B4%D0%B0-sql>
2. [https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS\\_SQLTUNE.html#GUID-58C2A9A0-A0E6-48FC-B69D-106C120E1A83](https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_SQLTUNE.html#GUID-58C2A9A0-A0E6-48FC-B69D-106C120E1A83)
3. [https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS\\_ADVISOR.html#GUID-905EA34C-6F2E-4AB0-9BA2-6DC612B4CF57](https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_ADVISOR.html#GUID-905EA34C-6F2E-4AB0-9BA2-6DC612B4CF57)

4. [https://docs.oracle.com/cd/B13789\\_01/server.101/b10755/statviews\\_2016.htm](https://docs.oracle.com/cd/B13789_01/server.101/b10755/statviews_2016.htm)
5. [http://citforum.ru/database/oracle/sqltune\\_add/](http://citforum.ru/database/oracle/sqltune_add/)
6. [https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS\\_ADVISOR.html#GUID-3E8DE78A-637F-477C-876E-C1F84A7C1602](https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_ADVISOR.html#GUID-3E8DE78A-637F-477C-876E-C1F84A7C1602)