

# 1 硬解析和软解析

说到软解析(soft parse)和硬解析(hard parse), 就不能不说一下Oracle对sql的处理过程。当你发出一条sql语句交付Oracle, 在执行和获取结果前, Oracle对此sql将进行几个步骤的处理过程:

## 1) 语法检查(syntax check)

检查此sql的拼写是否语法。

## 2) 语义检查(semantic check)

诸如检查sql语句中的访问对象是否存在及该用户是否具备相应的权限。

## 3) 对sql语句进行解析(parse)

利用内部算法对sql进行解析, 生成解析树(parse tree)及执行计划(execution plan)。

## 4) 执行sql, 返回结果(execute and return)

其中, 软、硬解析就发生在第三个过程里:

Oracle利用内部的hash算法来取得该sql的hash值, 然后在library cache里查找是否存在该hash值;

假设存在, 则将此sql与cache中的进行比较;

假设"相同", 就将利用已有的解析树与执行计划, 而省略了优化器的相关工作。这也就是软解析的过程。

诚然, 如果上面的2个假设中任有一个不成立, 那么优化器都将进行创建解析树、生成执行计划的动作。这个过程就叫硬解析。

创建解析树、生成执行计划对于sql的执行来说是开销昂贵的动作, 所以, 应当极力避免硬解析, 尽量使用软解析。

这就是在很多项目中, 倡导开发设计人员对功能相同的代码要努力保持代码的一致性, 以及要在程序中多使用绑定变量的原因。

# 2 不使用绑定变量 VS 使用绑定变量

## 2.1 不使用绑定变量

### 1) 创建表并测试

```
SQL> set timing on
```

```
SQL> create table t1(id int);
```

Table created.

Elapsed: 00:00:08.33

```
SQL> begin
  2 for i in 1 .. 100000
  3 loop
  4 execute immediate 'insert into t1 values('||i||)';
  5 end loop;
  6 commit;
  7 end;
  8 /
```

PL/SQL procedure successfully completed.

Elapsed: 00:01:45.70

```
SQL> set timing off
```

## 2) 查看硬解析次数以及执行次数

```
SQL> col sql_text for a40
```

```
SQL> col sql_id for a15
```

```
SQL> select sql_text,sql_id,executions,parse_calls from v$sql where sql_text like
'insert into t1 values%'and rownum <= 10;
```

SQL_TEXT	SQL_ID	EXECUTIONS	PARSE_CALLS
insert into t1 values(99985)	buvjmn4fvs5f3	1	1
insert into t1 values(99986)	2dm1pz9frh6bu	1	1
insert into t1 values(99959)	79v4n4vtnw8jc	1	1
insert into t1 values(99974)	5t392wysnh9hz	1	1
insert into t1 values(99993)	7c0bfs7vd89pt	1	1
insert into t1 values(99979)	17dpra9uznahb	1	1
insert into t1 values(99967)	9x4087b4rngh6	1	1

insert into t1 values(99965)	52a7wnjcy4qvz	1	1
insert into t1 values(99988)	9hcx3fv0q0r04	1	1
insert into t1 values(99989)	4ds5qvgv3hups	1	1

10 rows selected.

```
SQL> select count(*) from v$sql
  2 where sql_text like 'insert into t1 values%';
```

```
  COUNT(*)
```

```
-----
```

```
      49
```

思考：为何只有49次？

shared\_pool大小有限，无法保存每条解析过的sql，会通过LRU算法踢出冷块。

## 2.2 使用绑定变量

1) 清空缓存，测试

```
SQL> alter system flush shared_pool;
```

System altered.

```
SQL> alter system flush buffer_cache;
```

System altered.

```
SQL> drop table t1;
```

Table dropped.

```
SQL> set timing on
```

```
SQL> create table t1(id int);
```

Table created.

Elapsed: 00:00:00.23

```
SQL> begin
 2 for i in 1 .. 100000
 3 loop
 4 execute immediate 'insert into t1 values (:X)' using i;
 5 end loop;
 6 commit;
 7 end;
 8 /
```

PL/SQL procedure successfully completed.

Elapsed: 00:00:18.69

```
SQL> set timing off
```

2) 查看硬解析次数以及执行次数

```
SQL> select sql_text,sql_id,executions,parse_calls from v$sql where sql_text like
'insert into t1 values%';
```

SQL_TEXT	SQL_ID	EXECUTIONS	PARSE_CALLS
insert into t1 values (:X)	d1f3fv8rt9j8t	100000	1