Oracle nologging 何时生效

参考 ： http://www.eygle.com/faq/Nologging&append.htm

1. Nologging 的设置跟数据库的运行模式有关

a. 在非归档模式下

```
archive log list;
create table test as select * from dba_objects where 1=0;
select * from redo_size;
insert into test select * from dba_objects;
select * from redo_size;
insert /*+ append */ into test select * from dba_objects;
select * from redo_size;
drop table test;
```

可以看到这两次Insert 产生的日志量差距很大

b. 在归档模式下

```
shutdown immediate
startup mount
alter database archivelog;
alter database open;
archive log list;
create table test as select * from dba_objects where 1=0;
select * from redo_size;
insert into test select * from dba_objects;
select * from redo_size;
insert /*+ append */ into test select * from dba_objects;
select * from redo_size;
```

可以看到，在归档模式下，对于常规表的insert append 产生he insert 同样的redo. 此时的insert append 实际上并不会有性能提高。

通过 Logmnr 分析日志：

```
select operation, count(*) from v$logmnr_contents group by operation;
```

可以看到这里是direct insert, 且每条记录都产生的redo.

2. 对于 Nologging 的 table 的处理

a. 在归档模式下

```
create table test nologging as select * from dba_objects where 1=0;
select * from redo_size;
insert into test select * from dba_objects;
select * from redo_size;
insert /*+ append */ into test select * from dba_objects;
select * from redo_size;
```

可以看到，只有 append 才能减少redo

b. 在非归档模式下

```
shutdown immediate
startup mount
alter database noarchivelog;
alter database open;
create table test nologging as select * from dba_objects where 1=0;
select * from redo_size;
insert into test select * from dba_objects;
select * from redo_size;
insert /*+ append */ into test select * from dba_objects;
select * from redo_size;
```

同样，只有append 才能减少redo 的生成。

注：redo_size 是作者自己创建的一个View.

```
CREATE OR REPLACE VIEW redo_size
AS
SELECT VALUE
FROM v$mystat, v$statname
WHERE v$mystat.statistic# = v$statname.statistic#
AND v$statname.NAME = ’redo size’
/
```