

YUNHE

YUNHE ENMO (BEIJING) TECHNOLOGY CO.,LTD

Think Different

杨廷琨 (yangtingkun)

mail:tingkun.yang@enmotech.com

blog:http://yangtingkun.net

DTCC2012

个人介绍

□ 杨廷琨 (yangtingkun)

□ Oracle ACE

□ ITPUB数据库管理区版主

□ ACOUG核心会员

□ 参与编写《Oracle数据库性能优化》、
《Oracle DBA手记》和《Oracle DBA手记3》

□ 十二年的一线DBA经验

□ 个人BLOG中积累了2500篇原创技术文章

□ 云和恩墨资深技术经理



DTCC2012

Oracle能做什么

- Thomas Kyte: 在Oracle中，很少会说不能做什么，而是你会有多少种选择来实现这个功能。
- 我说：如果某个功能你在Oracle中无法实现，那么并不意味着在Oracle中无法实现，而多半是你对Oracle的功能还不是很了解。

我们能做什么

- Oracle数据库相关需求:

- 现有功能就能提供的超过60%。
- 现有功能进行简单封装的30%。
- 现有功能难以满足的不到10%。

初级DBA

中级DBA

高级DBA

Think Different

- 发挥创造力解决Oracle中难以解决的问题
- 想做的人会找个方法
- 不想做的人找个借口

Oracle难以实现的原因

- 数据库功能的限制条件
- 数据库版本的限制条件
- 数据库没有提供的功能

功能限制——LONG字段后添加字符

- 需求

查询LONG字段并添加字符串。

```
SELECT LONG_COL || '1234' FROM T;
```

- 问题

LONG类型无法使用||添加字符

功能限制——LONG字段后添加字符

解决方案

- PL/SQL实现 —— 处理长度受限
- 外部程序 —— 实现复杂度高
- 利用LOB进行转化

功能限制——LONG字段后添加字符

```
SQL> CREATE GLOBAL TEMPORARY TABLE T_LONG_LOB
2 (ID NUMBER, COL CLOB);
SQL> CREATE OR REPLACE FUNCTION F_LONG(P_ID IN NUMBER) RETURN
CLOB AS PRAGMA AUTONOMOUS_TRANSACTION;
2 V_RESULT CLOB;
3 BEGIN
4 INSERT INTO T_LONG_LOB SELECT ID, TO_LOB(COL)
5 FROM T_LONG WHERE ID = P_ID;
6 SELECT COL || '1234' INTO V_RESULT
7 FROM T_LONG_LOB WHERE ID = P_ID;
8 COMMIT;
9 RETURN V_RESULT;
10 END;
11 /
SQL> SELECT F_LONG(ID) FROM T_LONG;
```

功能限制——对LONG字段进行搜索

- 需求

对表中的LONG字段进行查询限定条件。

- 问题

- LONG字段类型字段无法出现在WHERE语句中
- LONG字段类型也无法创建索引

功能限制——对LONG字段进行搜索

- 解决方案
 - PL/SQL实现

功能限制——对LONG字段进行搜索

```
SQL> CREATE OR REPLACE FUNCTION F_QUERY_LONG (P_ID NUMBER,  
      P_STR VARCHAR2) RETURN NUMBER AS  
2  V_STR VARCHAR2(32767);  
3  BEGIN  
4  SELECT COL INTO V_STR FROM T_LONG WHERE ID = P_ID;  
5  IF INSTR(V_STR, P_STR) > 0 THEN RETURN 1;  
6  ELSE RETURN 0;  
7  END IF;  
8  END;  
9  /  
SQL> SELECT COL FROM T_LONG WHERE F_QUERY_LONG(ID, 'VIEW')  
      = 1;
```

功能限制——对LONG字段进行搜索

- 解决方案
 - PL/SQL实现
 - 有长度限制
 - 查询访问效率不高
 - 外部过程实现
 - 外部过程实现复杂度高
 - 查询访问效率不高
 - 全文索引

功能限制——对LONG字段进行搜索

```
SQL> CREATE INDEX IND_T_LONG_COL ON  
T_LONG (LONG_COL) INDEXTYPE IS CTXSYS.CONTEXT;
```

索引已创建。

```
SQL> SELECT ID FROM T_LONG WHERE  
CONTAINS (LONG_COL, 'WORLD') > 0;
```

ID

1

功能限制——对LONG字段进行搜索

- 解决方案
 - PL/SQL实现
 - 有长度限制
 - 查询访问效率不高
 - 外部过程实现
 - 外部过程实现复杂度高
 - 查询访问效率不高
 - 全文索引
 - 改变SQL写法，对应用不透明
 - 索引数据同步非实时

Oracle难以实现的原因

- 数据库功能的限制条件
- 数据库版本的限制条件
- 数据库没有提供的功能

版本限制——增加非空字段

- 需求

Oracle 10g给一个大表增加非空字段，减少对系统的影响，尽可能快的完成操作

版本限制——增加非空字段

- 解决方案
 - ALTER TABLE
 - 操作时间漫长
 - 对业务影响大
 - 行迁移问题
 - CREAT TABLE AS SELECT
 - 操作时间长
 - 对业务影响大
 - 在线重定义
 - 操作时间长

版本限制——增加非空字段

- 11g解决方案

```
ALTER TABLE T
```

```
ADD (COL VARCHAR2(30) DEFAULT 'A' NOT NULL);
```

- Pre 11g

优化的最高境界——DO NOTHING

版本限制——增加非空字段

- 解决方案
 - ALTER TABLE
 - 操作时间漫长
 - 对业务影响大
 - 行迁移问题
 - CREAT TABLE AS SELECT
 - 操作时间长
 - 对业务影响大
 - 在线重定义
 - 操作时间长
 - 视图封装

版本限制——增加非空字段

```
SQL> ALTER TABLE T_ADD_COLUMN ADD (NEW_COL VARCHAR2(30));
```

```
SQL> ALTER TABLE T_ADD_COLUMN MODIFY (NEW_COL DEFAULT 'OLD  
VALUE');
```

```
SQL> ALTER TABLE T_ADD_COLUMN RENAME TO T_ADD_COLUMN_BASE;
```

```
SQL> CREATE VIEW T_ADD_COLUMN  
2 (ID, NAME, NEW_COL)  
3 AS SELECT ID, NAME, NVL(NEW_COL, 'OLD VALUE')  
4 FROM T_ADD_COLUMN_BASE;
```

版本限制——增加非空字段

```
SQL> INSERT INTO T_ADD_COLUMN VALUES (12000, 'A', 'TEST');  
ORA-01733: virtual column not allowed here
```

```
SQL> CREATE OR REPLACE TRIGGER T_INS_I_TADDCOLUMN  
2  INSTEAD OF INSERT ON T_ADD_COLUMN  
3  BEGIN  
4  INSERT INTO T_ADD_COLUMN_BASE  
5  VALUES (:NEW.ID, :NEW.NAME, :NEW.NEW_COL);  
6  END;  
7  /
```

```
SQL> INSERT INTO T_ADD_COLUMN VALUES (12001, 'B', 'TEST');  
1 row created.
```

Oracle难以实现的原因

- 数据库功能的限制条件
- 数据库版本的限制条件
- 数据库没有提供的功能

无现有功能——读锁

- 需求

- 当一个会话访问这张表时，其他会话不能进行读取，而是等待访问的会话提交或回滚。
- 对应用透明，尽可能少修改程序
- 对所有客户端生效，而不是只针对程序

- 问题

- Oracle的锁粒度是读不阻塞写，写不阻塞读。
- 实现独占型读锁

无现有功能——读锁

- 解决方案
 - SELECT语句封装

无现有功能——读锁

```
SQL> CREATE OR REPLACE FUNCTION F_QUERY_T RETURN  
      SYS_REFCURSOR AS  
2  V_CURSOR SYS_REFCURSOR;  
3  BEGIN  
4  LOCK TABLE T IN EXCLUSIVE MODE;  
5  OPEN V_CURSOR FOR 'SELECT * FROM T';  
6  RETURN V_CURSOR;  
7  END;  
8  /  
SQL> SELECT F_QUERY_T FROM DUAL;
```

无现有功能——读锁

- 解决方案

- SELECT语句封装
 - 需要修改程序甚至需要修改访问方式，对用户不透明
 - 只能针对使用封装后的语句有效，对于直接访问的SQL无能为力
- DBMS_LOCK + VPD

无现有功能——读锁

● 读锁的实现——DBMS_LOCK

```
SQL> SELECT OBJECT_ID FROM USER_OBJECTS WHERE OBJECT_NAME =  
'T';
```

```
OBJECT_ID  
-----
```

```
93789
```

```
SQL> DECLARE
```

```
2 V_LOCK NUMBER;
```

```
3 BEGIN
```

```
4 V_LOCK := DBMS_LOCK.REQUEST(93789, RELEASE_ON_COMMIT =>  
    TRUE);
```

```
5 END;
```

```
6 /
```

无现有功能——读锁

- 锁的自动控制——VPD

```
SQL> CREATE OR REPLACE FUNCTION F_POLICY(OBJECT_SCHEMA IN  
VARCHAR2, OBJECT_NAME IN VARCHAR2)  
2 RETURN VARCHAR2 AS  
3 BEGIN  
4 RETURN ' DBMS_LOCK.REQUEST(93789, 6, 60) IN (0, 4) ' ;  
5 END;  
6 /
```

```
SQL> EXEC DBMS_RLS.ADD_POLICY(USER, ' T' , ' MYPOLICY' , USER,  
' F_POLICY' );
```

无现有功能——读锁

- 解决方案

- SELECT语句封装
 - 需要修改程序甚至需要修改访问方式，对用户不透明
 - 只能针对使用封装后的语句有效，对于直接访问的SQL无能为力
- DBMS_LOCK + VPD
 - SYS用户不受VPD策略影响

无现有功能——限制FOR UPDATE操作

- 需求

- 只给用户分配查询权限，而不给用户FOR UPDATE的能力。

- 问题

- 一旦分配SELECT权限，用户自动拥有FOR UPDATE能力。
- 用户虽然不能修改数据，但是可以锁定数据。

无现有功能——限制FOR UPDATE操作

- 解决方案
 - FIREWALL：通过直连的配置方式可以阻塞预配置好的FOR UPDATE操作
 - 实现复杂
 - 成本高
 - 只读
 - 影响大
 - 视图封装

无现有功能——限制FOR UPDATE操作

- 视图封装

```
SQL> create or replace view v_update as select rownum rn,  
a.* from t_update a;
```

```
SQL> select * from test.v_update where id = 1 for update;  
select * from test.v_update where id = 1 for update
```

ERROR at line 1:

ORA-02014: cannot select FOR UPDATE from view with DISTINCT,
GROUP BY, etc.

```
SQL> select id, name from test.v_update where id = 1 for  
update;
```

ID	NAME
----	------

1	a
---	---

无现有功能——限制FOR UPDATE操作

- 视图封装

```
SQL> create or replace view v_update as select distinct *  
      from t_update;
```

无现有功能——限制FOR UPDATE操作

- 视图封装

```
SQL> create or replace view v_update as
```

```
2 select * from t_update
```

```
3 union all
```

```
4 select * from t_update where 1 = 2;
```

```
SQL> select * from test.v_update where id = 1 for update;
```

```
select * from test.v_update where id = 1 for update
```

```
*
```

```
ERROR at line 1:
```

```
ORA-02014: cannot select FOR UPDATE from view with DISTINCT,  
GROUP BY, etc.
```

HOW TO?

- 尽可能熟悉Oracle的各种功能
- 大胆的假设、合理的推测、仔细的验证
- 百折不挠

Q&A



DTCC2012