vahidusefzadeh@gmail.com

# enq: TM - contention

enq: TM – contention معمولا به دو دلیل اصلی زیر رخ می دهد(البته دلایل دیگر هم دارد):

۱. عدم ایندکس گذاری بر روی کلید خارجی در جدول child

۲. direct load insert

این wait سبب می شود تا lock در سطح جدول ایجاد شود((table level lock(TM که منظور از TM همان table modification می باشد).

در این قسمت قصد داریم تا دو علت اصلی این wait را مورد بررسی قرار دهیم.

## unindexed foreign keys

```
drop table usef_child;
drop table usef_parent;

create table usef_parent(a number primary key,b number unique);
create table usef_child(c number,b_fk ,CONSTRAINT b FOREIGN KEY (b_fk) REFERENCES
usef_parent(b) ON DELETE CASCADE);

begin
for i in 1..10 loop
insert into usef_parent values(i+1,i+1);
commit;
end loop;
end;

insert into usef_parent values(10,10); commit;
```

**سناریوی اول:**

زمانی که چند کاربر قصد دارند رکوردهای جدول parent را حذف کنند که جدول child اجازه همزمانی این کار را نمی دهد:

| Session 1: | Session 2: |
|---|---|
| delete usef_parent where a=10<br>1 rows deleted(without commit); | |
| | delete usef_parent where a=6<br>waite…. |

با دو دستور زیر مشخص می شود که چه افرادی درگیر این نوع wait هستند:

```
select sid, sql_text from v$session s, v$sql q where sid in (select sid from v$session where state in  ('WAITING')
and wait_class != 'Idle' and event='enq: TM - contention' and (q.sql_id = s.sql_id  or q.sql_id = s.prev_sql_id));
```

vahidusefzadeh@gmail.com

| SID | SQL_TEXT |
|---|---|
| 129 | begin :id := sys.dbms_transaction.local_transaction_id; end; |
| 129 | delete usef_parent where a=6 |

select blocking_session, sid, serial#, wait_class, seconds_in_wait from v$session where blocking_session is not NULL order by blocking_session;

| BLOCKING_SESSION | SID | SERIAL# | WAIT_CLASS | SECONDS_IN_WAIT |
|---|---|---|---|---|
| 99 | 129 | 14 | Application | 814 |

**سناریوی دوم:**

فرض کنید نفر اول رکوردی را از جدول child حذف می کند ولی این عمل را commit یا rollback نکرده و در همین حال فردی می خواهد از جدول parent رکورد دیگری را حذف کند که در حالت انتظار قرار می گیرد.

| Session 1: | Session 2: |
|---|---|
| delete usef_child where b_fk=2<br>1 rows deleted(without commit); | |
| | delete usef_parent where a=10<br>waite…. |

راه حل:

همانطور که قبلا هم گفته شد، معمولا در سیستم ها دلیل اصلی رخ دادن این wait، به عدم ایندکس گذاری بر روی کلید خارجی بر می گردد و به همین دلیل توصیه می شود که برای همه کلیدهای خارجی، ایندکس گذاری صورت پذیرد(البته موارد استثنا را در ادامه ذکر خواهیم کرد).

create index usef_fk1 on USEF_CHILD(B_FK);

| Session 1: | Session 2: |
|---|---|
| **delete usef_child where b_fk=2**<br>**1 rows deleted(without commit);** | |
| | **delete usef_parent where a=10**<br>**1 rows deleted;** |

با استفاده از اسکریپت زیر، می توانیم تمامی کلیدهای خارجی ایندکس گذاری نشده را شناسایی کنیم:

**دانلود اسکریپت از سایت(http://vahidusefzadeh.blog.ir)**

**مزیت دیگر ایندکس گذاری کلید خارجی:**

زمانی که از ویژگی on delete cascade در هنگام ساخت کلید خارجی استفاده شود و در عین حال بر روی کلید خارجی ایندکسی موجود نباشد، باید یک عمل full-table scan بر روی جدول child صورت بگیرد تا رکورد در حال حذف شناسایی شود همچنین زمانی که از on delete restrict استفاده می شود، باز هم در صورت نبود ایندکس باید یک full table scan بر روی جدول child انجام بپذیرد.

ساخت ایندکس بر روی کلید خارجی سبب بهینه تر شدن عملیات join بین دو جدول parent و child هم می شود.

در صورتی که سه شرط زیر برقرار باشد، نیازی به ساخت ایندکس بر روی کلید خارجی نخواهیم داشت:

۱. مقادیر unique/primary key در جدول parent بروزرسانی نمی شوند.
۲. از جدول parent رکوردی حذف نمی شود.
۳. بین دو جدول parent و child، هیچگونه joinای رخ نمی دهد.

همچنین می توان TM lock را در سطح جدول غیرفعال کرد:

```
ALTER TABLE usef_child DISABLE TABLE LOCK;
```

| Session 1: | Session 2: |
| --- | --- |
| **delete usef_child where b_fk=2**<br>**1 rows deleted(without commit);** | |
| | **delete usef_parent where a=10**<br>**1 rows deleted;** |

با تنظیم پارامتر dml_locks به صفر، هر کسی که در این wait قرار بگیرد، کارش با خطا متوقف خواهد شد:

```
alter system set dml_locks=0 scope=spfile;
```

| Session 1: | Session 2: |
| --- | --- |
| delete usef_child where b_fk=2;<br>(without commit) | |
| | delete usef_parent where a=10;<br>ORA-00062: DML full table lock cannot be acquired; DML_LOCKS is 0 |

## direct load insert

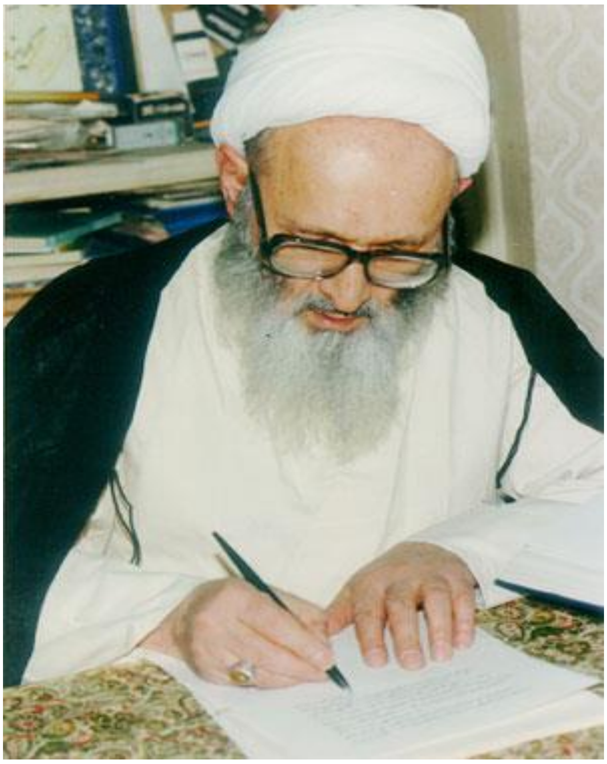
```
create table usef(a number,b number );

begin
for i in 1..10 loop
insert into usef values(i+1,i+1);
commit;
```

```
end loop;
end;
```

| Session 1: | Session 2: |
|---|---|
| insert **/*+ APPEND */** into usef select * from usef;<br>10 rows inserted(without commit); | |
| | insert **/*+ APPEND */** into usef select * from usef;<br>**wait….** |

vahidusefzadeh@gmail.com



علامه حسن زاده آملی:

**هرچه پخته تر شوی از نپخته ها، نپخته تر شنوی.**