



I. Tóm tắt bài thực hành

1. Yêu cầu lý thuyết

Sinh viên đã được trang bị kiến thức:

- Kiến thức về các phương thức khóa và các mức cô lập dữ liệu trong Oracle.
- Xây dựng các giao tác tường minh trên Oracle.

2. Nội dung

❖ Tìm hiểu về deadlock trong Oracle

- Khái niệm
- Phân loại

❖ Tìm hiểu về phương pháp kết nối Oracle với phần mềm

II. Deadlock

1. Khái niệm

Trong truy suất đồng thời, deadlock là một trạng thái trong đó các giao tác chờ nhau về mặt tài nguyên làm cho hệ thống đứng yên.

Ví dụ:

T1	T2	Ghi chú
Lock (A)		<i>T1 đợi T2 unlock(B) trong khi T2 lại đợi T1 unlock(A) để tiếp tục thực hiện. Hai giao dịch này cứ chờ nhau và cả hai đều không chạy được.</i>
	Lock (B)	
Lock (B) (Chờ)		
	Lock (A) (Chờ)	

2. Phân loại

Có hai loại deadlock sau:

❖ Cycle deadlock

T1	T2
begin tran (1) Write(A) DBMS_LOCK.SLEEP(15); (2) Write(B)	begin tran (3) Write(B)

commit	(4) Write(A) commit
--------	------------------------

Giả sử kịch bản của các giao tác trên là (1) → (3)

Nhận xét: T1 chờ T2 trả khóa ghi trên B và T2 chờ T1 trả khóa ghi trên A
→ Hệ thống bị treo.

❖ Conversion deadlock

T1	T2
begin tran SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; (1) Read(A) DBMS_LOCK.SLEEP(15); (2) Write(A) commit	begin tran SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; (3) Read(A) (4) Write(A) commit

Giả sử kịch bản của các giao tác trên là (1) → (3)

Nhận xét: T1 chờ T2 trả khóa đọc trên A và T2 chờ T1 trả khóa đọc trên A
→ Hệ thống bị treo.

Khi thực hiện từ hai transaction trở lên, một transaction sẽ bị hủy bỏ nếu nó yêu cầu lock mới và lock mới này có khả năng gây nên dealock. Sau đó, transaction còn lại sẽ được tiếp tục thực hiện cho đến khi hoàn tất.

Hệ quản trị cơ sở dữ liệu sẽ định kỳ kiểm tra deadlock. Nếu có deadlock, một trong những transaction phải bị từ bỏ để transaction kia tiếp tục. Để tránh Deadlock, transaction phải lấy được tất cả các khóa nó cần trước khi thực thi, giảm thiểu tối đa độ dài và thời gian thực thi của transaction.

III. Kết nối Oracle với phần mềm

Sinh viên tham khảo link video hướng dẫn trong file [KetNoiOracle.txt](#)

IV. Bài tập

1. Bài tập 1

1.1. Chuẩn bị

Cần mở hai cửa sổ SQL * Plus trên màn hình. Chúng ta sẽ giả sử rằng có hai người dùng, mỗi cửa sổ cho một người dùng. Đặt hai cửa sổ sao cho một cửa sổ ở phía trên màn hình và cửa sổ kia ở dưới màn hình. Đặt tên cửa sổ SQL * Plus trên là TOP và cửa sổ dưới là BOT. SET AUTOCOMMIT OFF ở cả hai cửa sổ.

1.2. Tạo bảng và dữ liệu

Tạo các bảng theo dõi số lượng hàng trong kho Các bộ phận của búp bê (DollParts) và Búp bê đã lắp ráp (Dolls). Thực hiện các câu lệnh SQL sau trong TOP hoặc BOT:

```
drop table DollParts;

create table DollParts (
    name varchar2(10) primary key,
    cnt number check (cnt >= 0));

insert into DollParts(name, cnt) values('HEAD', 17);
insert into DollParts(name, cnt) values('BODY', 17);
insert into DollParts(name, cnt) values('ARM', 17);
insert into DollParts(name, cnt) values('LEG', 17);
commit;

create table Dolls (
    name varchar2(20) primary key,
    cnt number);

insert into Dolls(name, cnt) values('Barbie', 0);
insert into Dolls(name, cnt) values('Ken', 0);
commit;
```

1.3. Thử tạo ra Deadlock

Bây giờ hãy chạy các câu lệnh SQL sau trong các cửa sổ TOP và BOT tương ứng:

```
TOP: update DollParts set cnt=100 where name='HEAD';
BOT: update Dolls set cnt=20 where name='Barbie';
TOP: update Dolls set cnt=10 where name='Barbie';
BOT: update DollParts set cnt=200 where name='HEAD';
```

Hoặc cấp quyền execute thủ tục dbms_lock cho user hiện hành và thực hiện đồng thời hai giao tác sau:

```
TOP:

begin
    update DollParts set cnt=100 where name='HEAD';
    dbms_lock.sleep(5);
```

```

        update Dolls set cnt=10 where name='Barbie';
    end;
/

BOT:

begin
    update Dolls set cnt=20 where name='Barbie';
    dbms_lock.sleep(5);
    update DollParts set cnt=200 where name='HEAD';
end;
/

```

Hãy trả lời các câu hỏi sau:

- Có tình trạng gì đã xảy ra?
- Hệ quản trị đã xử lý như thế nào?
- Giải thích.

1.4. Thử tạo ra Deadlock

Hãy chạy các câu lệnh SQL sau trong cửa sổ TOP và BOT tương ứng:

```

TOP:

declare
    dollName Dolls.name%TYPE := NULL;
    dollpartName DollParts.name%TYPE := NULL;
begin
    select name into dollName from Dolls where
name='Barbie' for UPDATE;
    dbms_lock.sleep(5);
    select name into dollpartName from DollParts where
name='HEAD' for UPDATE;
    rollback;
end;
/

BOT:

declare
    dollName Dolls.name%TYPE := NULL;
    dollpartName DollParts.name%TYPE := NULL;
begin
    select name into dollpartName from DollParts where
name='HEAD' for UPDATE;

```

```
dbms_lock.sleep(5);  
  
select name into dollName from Dolls where  
name='Barbie' for UPDATE;  
  
rollback;  
  
end;  
  
/
```

Hãy trả lời các câu hỏi sau:

- Có tình trạng gì đã xảy ra?
- Hệ quản trị đã xử lý như thế nào?
- Giải thích.

2. Bài tập 2

Yêu cầu sinh viên phát hiện các trường hợp có khả năng xảy ra deadlock trong cơ sở dữ liệu Quản lý thư viện và đề nghị cách giải quyết.

3. Bài tập 3

Sinh viên tìm hiểu phương pháp kết nối Oracle với ngôn ngữ mình đang sử dụng và viết báo cáo cụ thể từng bước thực hiện.

4. Bài tập 4

Yêu cầu sinh viên phát hiện tất cả các trường hợp có khả năng xảy ra deadlock trong đồ án môn học và đề nghị cách giải quyết.

~ HẾT ~