]

Database 2 Project

ORACLE TASK:

An organization has a "Department" table and an "Employees" table in Oracle. The "Department" table contains information about different departments in the organization, and the "Employees" table contains information about the employees, including the department they belong to. The department table contains ID as a primary key and department name. The departments are HR, IT, and finance. The Employee table contains ID as a primary key and name, salary, and department ID as a foreign key.

---

a) Create a Manager User and grant them the role of privileges to create two users. Let User 1 create the Employee and the Department table. Let User 2 insert 5 rows of employees. [2 Marks]

Answer:

- Create a Manager User

```
for admin:

sys / as sysdba
admin
create user amrr identified by 123;
grant create session to amrr with admin option;
grant create user to amrr;
grant create table to amrr with admin option;
```

- Grant them the role of privilege creating two users.

```
for user manager:
amrr
123
create user x identified by 123;
create user y identified by 123;
grant create session to y;
grant create session to x;
grant create table to x;

then

for admin:
alter user x quota 100m on system;
```

- Let User 1 create the Employee and the Department table

```
for user : x
x
123
create table department (
id INT PRIMARY KEY,
name VARCHAR(50)
);
```

```
CREATE TABLE employee8 (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    salary INT,
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES department(id)
);

grant insert on employee8 to y;
grant insert on department to y;
```

- Let User 2 insert 5 rows of employees.

```
for uesr :y
y
123
insert into x.department  values(1 ,'hr');
insert into x.department  values(2 ,'it');
insert into x.department  values(3 ,'finance');
commit;

insert into x.employee8 values(1,'amr',5000,1);
insert into x.employee8 values(2,'ali',6000,2);
insert into x.employee8 values(3,'yousef',7000,3);
insert into x.employee8 values(4,'mesho',8000,3);
insert into x.employee8 values(5,'mahmoud',9000,2);
commit;
```

---

b) Demonstrate generating a blocker-waiting situation using two transactions by user 1 and user 2. The Transaction is calling a function that raises the rate of salary by 10% for department 1. [2 Marks]

Answer:

```
for admin:
GRANT CREATE PROCEDURE TO x;
```

- Transaction user 1

```
for x:
BEGIN
    UPDATE x.employee8
    SET salary = salary * 1.1
    WHERE dept_id = 1;
END;

GRANT EXECUTE ON department1 TO y;

BEGIN
    x.department1;
    DBMS_OUTPUT.PUT_LINE('Procedure executed by user x');
END;
```

- Transaction user 2

```
for y:
BEGIN
    x.department1;
    DBMS_OUTPUT.PUT_LINE('Procedure executed by user y');
END;
```

---

c) Identify the sessions in the situation using SID and serial# for both blocker and waiting sessions. [2 Marks]

Answer:

```
for admin :
SELECT s1.sid AS blocker_sid, s1.serial# AS blocker_serial,
       s2.sid AS waiting_sid, s2.serial# AS waiting_serial
FROM V$SESSION s1, V$SESSION s2
WHERE s1.sid = (SELECT blocking_session FROM V$SESSION WHERE sid = s2.sid);
```

---

d) Demonstrate a deadlock scenario and display the expected result. [2 Marks]

Answer:

```
for x:
grant update on employee8 to y;
grant select on employee8 to y;
```

- Updates that will cause Deadlock :

```
for x:
UPDATE employee8 SET salary = 400 WHERE id = 2;

then
for y:
UPDATE x.employee8 SET salary = 900 WHERE id = 1;

then
for x:
UPDATE employee8 SET salary = 500 WHERE id = 1;

then
for y:
UPDATE x.employee8 SET salary = 800 WHERE id = 2;
```

- The expected results :
  (First we need to know the blacking and waiting sessions using this query)

```
admin:
SELECT s1.sid AS blocker_sid, s1.serial# AS blocker_serial,
       s2.sid AS waiting_sid, s2.serial# AS waiting_serial
FROM V$SESSION s1, V$SESSION s2
WHERE s1.sid = (SELECT blocking_session FROM V$SESSION WHERE sid = s2.sid)
```

After we know the blacking and waiting sessions ,We need to decide the victim session to kill

if you kill x session (blocking session) the results are:

```
Employee with ID = 1 has a salary of 900.
Employee with ID = 2 has a salary of 800.
```

if you kill y session (waiting session) the results are:

```
Employee with ID = 1 has a salary of 100.
Employee with ID = 2 has a salary of 400.
```

After we select the victim ,Now we will kill it :

```
ALTER SYSTEM KILL SESSION '14,147'; --'14,147' example of SID,SERAIL# of
killed session
```

---

e) Perform the following functions [2 Marks]

Answer:

- Create a function that calculates the average salary for any department

```
CREATE OR REPLACE FUNCTION avg_salary_for_dept(dept_id_in IN NUMBER)
RETURN NUMBER IS
  avg_sal NUMBER;
BEGIN
  SELECT AVG(salary)
  INTO avg_sal
  FROM employee8
  WHERE dept_id = dept_id_in;

  RETURN avg_sal;
END;
/

DECLARE
  avg_salary NUMBER;
BEGIN
  avg_salary := avg_salary_for_dept(1);
  DBMS_OUTPUT.PUT_LINE('Average salary for Department 1: ' || avg_salary);
END;
/
```

- Create a function that calculates the Total Salary in a Department.

```
CREATE OR REPLACE FUNCTION sum_salaries_for_dept(dept_id_in IN NUMBER)
RETURN NUMBER IS
  total_sal NUMBER;
BEGIN
  SELECT SUM(salary)
  INTO total_sal
  FROM employee8
  WHERE dept_id = dept_id_in;

  RETURN total_sal;
END;
/

SET SERVEROUTPUT ON;
DECLARE
  total_salary NUMBER;
BEGIN
  total_salary := sum_salaries_for_dept(2);
  DBMS_OUTPUT.PUT_LINE('Total salary for Department 2: ' || total_salary);
END;
/
```

- Create a function that calculates the maximum Salary.

```
CREATE OR REPLACE FUNCTION max_salary RETURN NUMBER IS
  max_sal NUMBER;
BEGIN
  SELECT MAX(salary)
  INTO max_sal
  FROM employee8;
  RETURN max_sal;
END;
/

SET SERVEROUTPUT ON; -- Enable DBMS_OUTPUT
SET SERVEROUTPUT ON SIZE UNLIMITED;

BEGIN

  DBMS_OUTPUT.PUT_LINE('Maximum salary across all departments: ' ||
max_salary);
END;
/
```