**Written in ADMIN Connection(1):**

**--Creating Manager User and Role**

create user manager identified by 123;

create role mngRole;

**--Granting privs to the role then granting the role to the manager user**

grant create session, create user,create table, create procedure, unlimited tablespace to mngRole with admin option;

grant mngRole to manager;

**Written in MANAGER Connection(2):**

**--Creating both users and granting them privs**

create user user1 identified by 123;

grant create session,create table, unlimited tablespace to user1;

create user user2 identified by 123;

grant create session,unlimited tablespace to user2;

grant create procedure to user1,user2;

**Written in USER1 Connection(3):**

**--Creating Department Table**

create table department(

id int primary key,

name varchar(50)

);

**--Creating Employee Table**

create table employee(

id int primary key,

name varchar(50),

salary int,

depID int,

CONSTRAINT departmentID\_fk foreign key (depID) references department(id)

);

--**Inserting Department Data into Department Table**

insert into user1.department values(1,'HR');

insert into user1.department values(2,'IT');

insert into user1.department values(3,'Finance');

**--Granting Privs to User2**

grant insert,delete,update,select on department to user2;

grant insert,delete,update,select on employee to user2;

**Written in USER2 Connection(4):**

**--Inserting 5 rows in the EMPLOYEE table**

insert into user1.employee values(1,'Ahmed',10000,1);

insert into user1.employee values(2,'Mohamed',8000,2);

insert into user1.employee values(3,'Moaz',5000,3);

insert into user1.employee values(4,'Salma',10,1);

insert into user1.employee values(5,'Gasser',6700,3);

**Written in USER1 Connection(5):**

**--Creating the function that updates the salary**

CREATE OR REPLACE FUNCTION updatedSal(deptID number)

return number as

begin

IF (deptID = 1)

THEN

update employee set salary = salary + salary \*.1

where depID = deptID;

return SQLCODE;

end IF;

end updatedSal;

/

**--Granting execution to USER2**

grant execute on updatedSal to user2;

**Written in BOTH USERS’ Connection(6):**

**--Executing the function to generate the BLOCKER-WAITING Situation**

begin

dbms\_output.put\_line(updatedSal(1));

end;

/

**Written in ADMIN Connection(7):**

**--Selecting the SID AN S# of the blocker and the waiter after generating the situation**

select

blocker.sid as blockerSID,

blocker.serial# as blockerSerial,

waiter.sid as waiterSID,

waiter.serial# as waiter\_serial

from

v$session blocker

join v$session waiter on blocker.sid = waiter.blocking\_session

where waiter.blocking\_session is not null;

**Written in BOTH USERS’ Connection(8):**

**--Updating both employees data to generate a DEADLOCK situation**

update user1.Employee set salary = salary + salary \* 0.1 where user1.Employee.ID = 1;

update user1.Employee set salary = salary + salary \* 0.1 where user1.Employee.ID = 2;

**Written in USER1 Connection(9):**

**--Function that calculates the average Salary of any department**

CREATE OR REPLACE FUNCTION avgSalary(deptID number)

return number as avgSal number;

begin

select avg(salary) into avgSal from user1.employee where depID = deptID;

return avgSal;

end avgSalary;

/

**--Executing it**

select distinct(avgSalary(2)) from employee;

**--Function that calculates the total salary in a Department**

CREATE OR REPLACE FUNCTION sumSalary(deptID number)

return number as sumSal number;

begin

select sum(salary) into sumSal from user1.employee where depID = deptID;

return sumSal;

end sumSalary;

/

**--Executing it**

select distinct(sumSalary(1)) from employee;

--**A function that calculates max salary**

CREATE OR REPLACE FUNCTION maxSalary

return number as maxSal number;

begin

select max(salary) into maxSal from user1.employee;

return maxSal;

end maxSalary;

/

**--Executing it**

select distinct(maxSalary) from employee;

**--Granting USER2 Execution of all functions**

grant execute on avgSalary to user2;

grant execute on sumSalary to user2;

grant execute on maxSalary to user2;