

## Database and Information Systems 2020/2021, Test 1 (SQL SELECT), Variant: A

Duration: 70min

Maximal number of points/minimal number of points: 15/8

Only complete and correct solution will be evaluated by non-zero points.

---

### Tasks

1. Create procedure with input parameters `p_deviceID`, `p_eventTypeID`, `p_personID`, `p_startDate`, `p_endDate` that will insert event into table `device_event`. Firstly, the procedure will verify whether the start date is older than the end date and if not, it displays a message on the screen: "The start date cannot be newer than the end date !!!" and the procedure will end. Subsequently, the procedure will verify whether the employee with the given ID exists and if not, it displays the message on the screen: "The employee with the given ID does not exist !!!" and the procedure will end. Finally, the procedure will verify whether the employee with the given ID is an ordinary employee or a boss. For an ordinary employee, the event is inserted in the table. For the boss, the procedure will insert the event by assigning it to the ordinary employee of the boss with the largest number of assigned events. In both cases, the procedure will display on the screen to whom the event was assigned (his name). The task will be handled as a transaction.

```
create or replace PROCEDURE InsertDeviceEvent
(
    p_deviceID IN device_event.dID%type,
    p_eventTypeID IN device_event.tID%type,
    p_personID IN device_event.pID%type,
    p_startDate IN device_event.startDate%type,
    p_endDate IN device_event.endDate%type
)
AS
    v_exists INT;
    v_boss INT;
    v_name person.name%type;
    v_employeeID int;
    v_eID int;
BEGIN
    if p_startDate > p_endDate then
        DBMS_OUTPUT.PUT_LINE('The start date cannot be newer than the end date !!!');
    else
        SELECT count(*) into v_exists from person where pID = p_personID;
        if v_exists = 0 then
            DBMS_OUTPUT.PUT_LINE('The employee with the given ID does not exist !!!');
        else
            SELECT max(eID) + 1 into v_eID FROM device_event;
            SELECT bossID into v_boss from person where pID = p_personID;
            if v_boss IS NOT NULL then
                INSERT INTO device_event (eID,dID,pID,tID,startDate,endDate)
                VALUES (v_eID, p_deviceID, p_personID, p_eventTypeID, p_startDate, p_endDate);

                SELECT name into v_name FROM person WHERE pID = p_personID;
                DBMS_OUTPUT.PUT_LINE('Event assigned to ' || v_name);
            else
                SELECT person.pID into v_employeeID
                FROM device_event JOIN person on device_event.pID = person.pID
                WHERE bossID = p_personID
                GROUP BY person.pID
                ORDER BY count(eID) DESC
                FETCH FIRST 1 ROWS ONLY;

                INSERT INTO device_event (eID,dID,pID,tID,startDate,endDate)
                VALUES(v_eID, p_deviceID, v_employeeID, p_eventTypeID, p_startDate, p_endDate);

                SELECT name into v_name FROM person WHERE pID = v_employeeID;
                DBMS_OUTPUT.PUT_LINE('Event assigned to ' || v_name);
            end if;
        end if;
    end if;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('error');
        ROLLBACK;
END;
```

2. Create table Statistic with the attributes sID primary key autoincremented integer, pID integer, count integer, order integer.

Create a procedure with one input parameter p\_year which will count how many events in the given year were assigned to the employee and inserts these statistics into the table Statistics. Except pID of the employee and the number of events the record will contain also the order (attribute order) set according to the number of events (i.e. the employee with the highest number of events will have the order 1, etc.).

```
CREATE TABLE Statistics
(
    sID int GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),
    pID int not null,
    scount int not null,
    sorder int not null,
    PRIMARY KEY (sID)
);

CREATE OR REPLACE PROCEDURE CreateStatistics (p_year IN INT)
AS
    CURSOR employees is SELECT pID, count(eID) as cnt
                        FROM device_event
                        WHERE EXTRACT(year from startdate) = p_year
                        GROUP BY pID
                        ORDER BY cnt DESC;

    v_counter int;
BEGIN
    v_counter:=1;
    for employee in employees loop
        INSERT INTO Statistics(pID,scount,sorder) VALUES(employee.pID, employee.cnt, v_counter);
        v_counter := v_counter + 1;
    end loop;
END;
```