

Homework 2 – Mensio Martino 191854

The triggers defined below can manage also situations where a phone moves and changes cell. Switching on a phone is not a problem because there is only the current position of a phone. Instead when turning off a phone, I need to update the cell where the phone was known to be previously. Also in the case of the phone call start, I need to update the attributes of the current cell from where the phone call wants to start and also the previous cell (where the phone was turned on and now is no more present).

In the case phones never change their position, the triggers lead the database in the same final state, performing a bit slower and accessing more tuples and tables, compared to the triggers I would have written knowing that the position was fixed.

1. Switching on/off the phone

- Triggers definition:

```
-- trigger for managing phones turning on
create or replace trigger active_Phone
after insert on state_change
for each row
when (NEW.changeType = 'O')
declare
    myCellId number;
begin
    -- find the cell where the phone is located. The max is for error detection
    select max(cellId) into myCellId
    from cell
    where x0 <= :NEW.x and x1 > :NEW.x and y0 <= :NEW.y and y1 > :NEW.y;

    -- in case no cell covers the location of the phone, raise an application error
    if (myCellId IS NULL) then
        raise_application_error(-20500, 'No cell found in this region of space');
    end if;

    -- insert the telephone in the table
    insert into telephone(phoneNo, x, y, phoneState)
    values(:NEW.phoneNo, :NEW.x, :NEW.y, 'On');

    -- update the current number of phones in the cell
    update cell
    set currentPhone# = currentPhone# + 1
    where cellId = myCellId;
end;
/

-- trigger for managing phones turning off
create or replace trigger off_Phone
after insert on state_change
for each row
when (NEW.changeType = 'F')
declare
    myPhoneLastX number;
    myPhoneLastY number;
    myCellId number;
begin
    -- find the last known location for the phone
    select x, y into myPhoneLastX, myPhoneLastY
    from telephone
    where phoneNo = :NEW.phoneNo;

    -- find the cell where the phone is located. The max is for error detection.
    -- If I assume that the location of a phone does not change, I could simply
    -- find the cellId with the following query (no need to retrieve last known position)
    -- select max(cellId) into myCellId
    -- from cell
```

```

-- where x0 <= :NEW.x and x1 > :NEW.x and y0 <= :NEW.y and y1 > :NEW.y;
-- but instead if i switch off a phone from a different location, i need to update the
-- currentPhone# from the cell where the phone was previously (and where it has been counted)
select max(cellId) into myCellId
from cell
where x0 <= myPhoneLastX and x1 > myPhoneLastX and y0 <= myPhoneLastY and y1 > myPhoneLastY;

-- in case no cell covers the location of the phone, raise an application error
if (myCellId IS NULL) then
    raise_application_error(-20500, 'No cell found in this region of space');
end if;

-- delete the telephone from the table
delete from telephone
where phoneNo = :NEW.phoneNo;

-- update the current number of phones in the cell
update cell
set currentPhone# = currentPhone# - 1
where cellId = myCellId;
end;

```

- Trigger behavior verification

```
select * from telephone; select * from cell;
```

nessuna riga selezionata

CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	0	3
2	10	0	20	10	0	20
3	0	10	10	20	0	20
4	10	10	20	20	0	20

```

-- active_Phone test
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (1, sysdate, '333000010', 3, 3, 'O');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (2, sysdate, '333000009', 15, 15, 'O');
select * from telephone; select * from cell;

```

1 righe inserito.

1 righe inserito.

PHONENO	X	Y	PHONESTATE
333000010	3	3	On
333000009	15	15	On

CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	1	3
2	10	0	20	10	0	20
3	0	10	10	20	0	20
4	10	10	20	20	1	20

```

-- off_Phone test
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (3, sysdate, '333000009', 15, 15, 'F');
select * from telephone; select * from cell;

```

1 righe inserito.						
PHONENO	X	Y PHONESTATE				
333000010	3	3 On				
CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	1	3
2	10	0	20	10	0	20
3	0	10	10	20	0	20
4	10	10	20	20	0	20

2. Starting a phone call

- Trigger definition

This trigger manages also cases where a call starts from a position different from the position where the phone was turned on. In this case, the trigger updates the `currentPhone#` attribute in the table CELL for the previous cell and the current cell. The trigger updates also the position of the phone in the TELEPHONE table, so that following trigger executions will be able to calculate the right number of phones calling that use a specific cell.

```
-- trigger for managing call start
create or replace trigger start_Call
after insert on state_change
for each row
when (NEW.changeType = 'C')
declare
  myPreviousCellId number;
  myCellId number;
  myMaxCalls number;
  myCellx0 number;
  myCellx1 number;
  myCelly0 number;
  myCelly1 number;
  myActivePhones# number;
  myMaxExId number;
begin
  -- if the phone starts a call in a different cell from where it was turned on, need to update
  -- the telephone table and the value of activePhone# of the previous and current cell
  select c.cellId into myPreviousCellId
  from telephone t, cell c
  where t.phoneNo = :NEW.phoneNo
  and c.x0 <= t.x and c.x1 > t.x and c.y0 <= t.y and c.y1 > t.y;

  -- retrieve some information of the cell that covers the phone
  select max(cellId), max(maxCalls), max(x0), max(x1), max(y0), max(y1)
  into myCellId, myMaxCalls, myCellx0, myCellx1, myCelly0, myCelly1
  from cell
  where x0 <= :NEW.x and x1 > :NEW.x and y0 <= :NEW.y and y1 > :NEW.y;

  -- let's update the phone position inside the telephone table (If it does not change, who cares)
  update telephone
  set x = :NEW.x, y = :NEW.y
  where phoneNo = :NEW.phoneNo;

  -- let's see if the phone changed cell
  if (myCellId <> myPreviousCellId) then
    update cell
    set currentPhone# = currentPhone# + 1
    where cellId = myCellId;

    update cell
    set currentPhone# = currentPhone# - 1
    where cellId = myPreviousCellId;
```

```

end if;

-- in case no cell covers the location of the phone, raise an application error
if (myCellId IS NULL) then
    raise_application_error(-20500, 'No cell found in this region of space');
end if;

-- find out how many phones are active (calling) covered by the cell
select count(*) into myActivePhones#
from telephone
where phoneState = 'Active' and x >= myCellx0 and x < myCellx1 and y >= myCelly0 and y < myCelly1;

-- compare the number of active phones in the cell with the cell limit
if (myActivePhones# >= myMaxCalls) then
    -- in case the call can't be served, find the id of the last exception occurred
    select max(exId) into myMaxExId
    from exception_log
    where cellId = myCellId;

    -- if no exceptions have occurred, the id should be 1
    if (myMaxExId IS NULL) then
        myMaxExId := 0;
    end if;

    -- insert the new exception
    insert into exception_log(exId, cellId, exceptionType)
    values(myMaxExId + 1, myCellId, 'N');
else
    -- if the phone can be served, change the phone state
    update telephone
    set phoneState = 'Active'
    where phoneNo = :NEW.phoneNo;
end if;
end;

```

- **Trigger behavior verification**

The starting database state is the result of the previous point.

```

-- start_Call test
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (4, sysdate, '333000001', 3, 3, 'O');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (6, sysdate, '333000004', 5, 5, 'O');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (7, sysdate, '333000004', 5, 5, 'C');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (8, sysdate, '333000001', 3, 3, 'C');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (9, sysdate, '333000010', 3, 3, 'C');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (10, sysdate, '333000020', 4, 4, 'O');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (11, sysdate, '333000020', 4, 4, 'C');
select * from telephone; select * from cell; select * from exception_log;

```

1 righe inserito.
 1 righe inserito.
 1 righe inserito.
 1 righe inserito.
 1 righe inserito.
 1 righe inserito.
 1 righe inserito.

• Cattura rettangolare

PHONENO	X	Y	PHONESTATE
333000010	3	3	Active
333000001	3	3	Active
333000004	5	5	Active
333000020	4	4	On

CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	4	3
2	10	0	20	10	0	20
3	0	10	10	20	0	20
4	10	10	20	20	0	20

EXID	CELLID	EXCEPTIONTYPE
1	1	N

• Extra verifications

In addition, to test the correct behavior of the trigger when the position of a phone changes, I performed some other operation on the database.

```
update cell
set maxcalls = 2
where cellid = 3;
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (12, sysdate, '333000020', 4, 15, 'C');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (13, sysdate, '333000004', 4, 15, 'C');
INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
values (14, sysdate, '333000001', 4, 15, 'C');
select * from telephone; select * from cell; select * from exception_log;
```

1 righe aggiornato.
 1 righe inserito.
 1 righe inserito.
 1 righe inserito.

PHONENO	X	Y	PHONESTATE
333000010	3	3	Active
333000001	4	15	Active
333000004	4	15	Active
333000020	4	15	Active

CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	1	3
2	10	0	20	10	0	20
3	0	10	10	20	3	2
4	10	10	20	20	0	20

EXID	CELLID	EXCEPTIONTYPE
1	1	N
1	3	N
2	3	N

3. Changing the maximum number of active calls

- Trigger definition

```
-- trigger for managing updates on maxCalls (is a before trigger because it must perform compensative action)
create or replace trigger change_MaxCalls
before update of maxCalls on cell
for each row
declare
  myActivePhones# number;
begin
  -- find out how many active phones are using this cell
  select count(*) into myActivePhones#
  from telephone
  where phoneState = 'Active' and x >= :NEW.x0 and x < :NEW.x1 and y >= :NEW.y0 and y < :NEW.y1;

  -- if there are more active phones than the new value, enforce the constraint
  if (myActivePhones# > :NEW.maxCalls) then
    -- prevent the wrong update
    :NEW.maxCalls := myActivePhones#;
  end if;
end;
```

- Trigger behavior verification

```
-- change_MaxCalls test
select * from cell; select * from telephone;
```

CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	4	3
2	10	0	20	10	0	20
3	0	10	10	20	0	20
4	10	10	20	20	0	20

PHONENO	X	Y	PHONESTATE
333000010	3	3	Active
333000001	3	3	Active
333000004	5	5	Active
333000020	4	4	On

```
UPDATE CELL SET MaxCalls = MaxCalls-2;
select * from cell;
```

4 righe aggiornato.

CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	4	3
2	10	0	20	10	0	18
3	0	10	10	20	0	18
4	10	10	20	20	0	18

4. Service guarantee

- Trigger definition

```
create or replace trigger service_Guarantee
after update on CELL
declare
    myTotalMaxCalls number;
begin
    -- calculate the maximum number of active calls considering all the cells
    select sum(maxCalls) into myTotalMaxCalls
    from cell;

    -- if it is lower or equal to 30, raise application error
    if (myTotalMaxCalls <= 30) then
        raise_application_error(-20500, 'Service guarantee constraint not satisfied. The total maximum number of
calls must be grater than 30');
    end if;
end;
```

- Trigger behavior verification

The starting database state is the result of the previous point.

```
-- service_Guarantee constraint test
UPDATE CELL SET MaxCalls = MaxCalls-1;
UPDATE CELL SET MaxCalls = MaxCalls-10;
select * from cell;
```

4 righe aggiornato.

Errore con inizio alla riga : 34 nel comando -

UPDATE CELL SET MaxCalls = MaxCalls-10

Report error -

Errore SQL: ORA-20500: Service guarantee constraint not satisfied. The total maximum number of calls must be grater than 30

ORA-06512: at "MARTINO.SERVICE_GUARANTEE", line 10

ORA-04088: error during execution of trigger 'MARTINO.SERVICE_GUARANTEE'

CELLID	X0	Y0	X1	Y1	CURRENTPHONE#	MAXCALLS
1	0	0	10	10	4	3
2	10	0	20	10	0	17
3	0	10	10	20	0	17
4	10	10	20	20	0	17