

# Database Management Systems

## Homework 2

The following relations are given (primary keys are underlined, optional attributes are denoted with \*):

```
CELL(CellId, x0, y0, x1, y1, CurrentPhone#, MaxCalls)
TELEPHONE(PhoneNo, x, y, PhoneState)
STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType)
EXCEPTION_LOG(ExId, CellId, ExceptionType)
```

The schema describes a cellular phone network. Telephones in the network are described by the **TELEPHONE** table. The phone state may take the values **On** or **Active** (the phone is **Active** if a phone call is taking place). When the phone is off, it is not present in the **TELEPHONE** table. The current position of each phone in the network is given by its **x, y** coordinates.

The cellular phone network is formed by cells. For the sake of simplicity, square cells are considered. Cells in the network are stored in the **CELLS** table. Each cell position in the network is defined by the **x<sub>0</sub>, y<sub>0</sub>** coordinates of its inferior left vertex and by the **x<sub>1</sub>, y<sub>1</sub>** coordinates of its superior right vertex. A phone belongs to the single cell satisfying both conditions  $x_0 \leq x < x_1$  e  $y_0 \leq y < y_1$ . For each cell, the number of phones currently in the cell (**CurrentPhone#** attribute) and the maximum number of active calls (**MaxCalls** attribute) are stored.

Write the triggers managing the following network state changes (insertions in the **STATE-CHANGE** table).

*1) Switching on/off the phone.* The change types are 'O' (on) and 'F' (off). When the phone is switched on, the corresponding information is stored in the **TELEPHONE** table. When it is switched off, the information should be removed. Furthermore, the cell to which the phone belongs should be identified and the current number of phones should be modified accordingly.

*2) Starting a phone call.* The change type is 'C'. If the cell in which the phone is located does not exceed the maximum number of calls it can manage (**MaxCalls** attribute), the phone state should become 'Active'. If instead the cell exceeds the maximum call number, the phone call cannot be initiated. In this case, the information on the exception should be inserted in the **EXCEPTION-LOG** table. The **ExId** attribute is a counter, which is unique for a given cell.

Write the triggers managing updates on the maximum number of active calls (updates on **MaxCalls** attribute in the **CELL** table).

*3) Changing the maximum number of active calls.* The maximum number of active calls related to a single cell may be reduced by the cellular phone network for managing issues (decrease of the **MaxCalls** value in the **CELL** table). The update on the **MaxCalls** attribute for a single cell could cause an inconsistent situation in which the **MaxCalls** value in the **CELL** table becomes smaller than the number of currently **Active** phones (**PhoneState='Active'**) in the considered cell. If so, the corresponding **MaxCalls** attribute needs to be updated with the number of currently **Active** phones (**PhoneState='Active'**) in the considered cell.

*4) Service guarantee.* The cellular phone network administrator needs to guarantee a minimum level service. In particular, the maximum number of active calls, by considering all cells of the network, needs to be always greater than 30. Thus, updates on the **MaxCalls** attribute in the **CELL** table must always satisfy the constraint. When an update instruction on **MaxCalls** attribute in the **CELL** table does not satisfy this constraint, the trigger will raise an application error to disallow the instruction that activates the trigger.

## Procedure:

- Create the database using the `create_db_telefoni_en.sql`
- Create the trigger according to the requirements stated at step 1
- Check that the trigger (*Active\_Phone*) works by inserting two records in the **STATE\_CHANGE** table:
  - `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');`

- Check that the trigger (*Off\_Phone*) works by inserting the following record in the **STATE\_CHANGE** table:
  - `INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType) values (3, sysdate, '333000009', 15, 15, 'F');`
- Create the trigger according to the requirements stated at step 2
- Check that the trigger works by inserting the following records in the **STATE\_CHANGE** table:
  - `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');`
- Create the trigger according to the requirements stated at step 3
- Check that the trigger works by running the following update:
  - `UPDATE CELL SET MaxCalls = MaxCalls-2;`
- Create the trigger according to the requirements stated at step 4
- Check that the trigger works by running the following updates:
  - `UPDATE CELL SET MaxCalls = MaxCalls-1;`
  - `UPDATE CELL SET MaxCalls = MaxCalls-10;`

**In the solution of the homework, for all triggers**

- **Report the text of the trigger**
- **Verify the database state at each step, by checking database modifications**