

Quaderno #3- Trigger

D'Andrea Giuseppe

Punto 1)

- *Evento*: INSERT ON STATE_CHANGE
- *Semantica*: AFTER / FOR EACH ROW
- *Condizione*: NEW.ChangeType = 'O' OR NEW.ChangeType = 'F'
- *Azione*:
 1. Trovare la cella di appartenenza del telefono
 2. Se il telefono viene acceso, inserire le informazioni su di esso nella tabella TELEPHONE ed incrementare il conteggio dei telefoni presenti nella cella
 3. Se il telefono viene spento, eliminare le informazioni su di esso nella tabella TELEPHONE e decrementare il conteggio dei telefoni presenti nella cella

Trigger

```
CREATE OR REPLACE TRIGGER OnOffTelephone
AFTER INSERT ON STATE_CHANGE
FOR EACH ROW
WHEN (NEW.ChangeType = 'O' OR NEW.ChangeType = 'F')
DECLARE
Cell INTEGER;
BEGIN
    -- Trovare la cella di appartenenza del telefono
    SELECT CellId INTO Cell
    FROM CELL
    WHERE :NEW.x >= x0 AND :NEW.x < x1 AND :NEW.y >= y0 AND :NEW.y < y1;
    IF (:NEW.ChangeType = 'O') THEN
        -- Se il telefono viene acceso, inserire le informazioni su di esso nella tabella
        -- TELEPHONE ed incrementare il conteggio dei telefoni presenti nella cella
        INSERT INTO TELEPHONE(PhoneNo, x, y, PhoneState)
        VALUES(:NEW.PhoneNo, :NEW.x, :NEW.y, 'On');
        UPDATE CELL
        SET CurrentPhone# = CurrentPhone# + 1
        WHERE CellId = Cell;
    END IF;
    IF (:NEW.ChangeType = 'F') THEN
        -- Se il telefono viene spento, eliminare le informazioni su di esso nella tabella
        -- TELEPHONE e decrementare il conteggio dei telefoni presenti nella cella
        DELETE FROM TELEPHONE
        WHERE PhoneNo = :NEW.PhoneNo;
        UPDATE CELL
        SET CurrentPhone# = CurrentPhone# - 1
        WHERE CellId = Cell;
    END IF;
END;
```

Queries

```
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');
INSERT INTO STATE_CHANGE(ChangeId , TimeStamp , PhoneNo , x, y, ChangeType)
VALUES (3, sysdate , '333000009', 15, 15, 'F');
```

Screenshots

CHANGEID	TIMEST...	PHONENO	X	Y	CHANGE...
----------	-----------	---------	---	---	-----------

Figura 1 - Tabella TELEPHONE prima delle query

	CHANGEID	TIMESTAMP	PHONENO	X	Y	CHANGETYPE
1		1 07-DIC-20	333000010	3	3	O
2		2 07-DIC-20	333000009	15	15	O
3		3 07-DIC-20	333000009	15	15	F

Figura 2 - Tabella TELEPHONE dopo le query

PHONENO	X	Y	PHONES...
---------	---	---	-----------

Figura 3 - Tabella STATE_CHANGE prima delle query

	PHONENO	X	Y	PHONESTATE
1	333000010	3	3	On

Figura 4 - Tabella STATE_CHANGE dopo le query

Punto 2)

- *Evento:* INSERT ON STATE_CHANGE
- *Semantica:* AFTER / FOR EACH ROW
- *Condizione:* NEW.ChangeType = 'C'
- *Azione:*
 1. Trovare la cella di appartenenza del telefono e salvare le relative coordinate e il numero massimo di telefoni che possono effettuare contemporaneamente una chiamata
 2. Calcolare il numero di chiamate attive nella cella di appartenenza del telefono
 3. Se la cella non supera il numero massimo di chiamate che può gestire, modificare lo stato del telefono su 'Active' nella tabella TELEPHONE
 4. Se la cella supera il numero massimo di chiamate che può gestire, inserire l'informazione nella tabella EXCEPTION_LOG dopo aver trovato il valore massimo del contatore per la cella considerata

Trigger

```
CREATE OR REPLACE TRIGGER TelephoneCall
AFTER INSERT ON STATE_CHANGE
FOR EACH ROW
WHEN (NEW.ChangeType = 'C')
DECLARE
Cell INTEGER;
x0 DECIMAL(7,2);
x1 DECIMAL(7,2);
y0 DECIMAL(7,2);
y1 DECIMAL(7,2);
MaxCalls SMALLINT;
Calls SMALLINT;
MaxExId INTEGER;
BEGIN
-- Trovare la cella di appartenenza del telefono e salvare le relative coordinate e il numero
-- massimo di telefoni che possono effettuare contemporaneamente una chiamata
SELECT CellId, x0, x1, y0, y1, MaxCalls INTO Cell, x0, x1, y0, y1, MaxCalls
FROM CELL
WHERE :NEW.x >= x0 AND :NEW.x < x1 AND :NEW.y >= y0 AND :NEW.y < y1;
-- Calcolare il numero di chiamate attive nella cella di appartenenza del telefono
SELECT COUNT(*) INTO Calls
FROM TELEPHONE
WHERE PhoneState = 'Active' AND x >= x0 AND x < x1 AND y >= y0 AND y < y1;
IF (Calls + 1 <= MaxCalls) THEN
-- Se la cella non supera il numero massimo di chiamate che può gestire, modificare lo
-- stato del telefono su 'Active' nella tabella TELEPHONE
UPDATE TELEPHONE
SET PhoneState = 'Active'
WHERE PhoneNo = :NEW.PhoneNo;
ELSE
-- Se la cella supera il numero massimo di chiamate che può gestire, inserire
-- l'informazione nella tabella EXCEPTION_LOG dopo aver trovato il valore massimo del
-- contatore per la cella considerata
SELECT MAX(ExId) INTO MaxExId
FROM EXCEPTION_LOG
WHERE CellId = Cell;
IF (MaxExId IS NULL) THEN
MaxExId := 0;
END IF;
INSERT INTO EXCEPTION_LOG(ExId, CellId, ExceptionType)
VALUES (MaxExId + 1, Cell, 'M');
END IF;
END;
```

Queries

```
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 (5, sysdate , '333000004', 5, 5, 'O');
INSERT INTO STATE_CHANGE(ChangeId , TimeStamp , PhoneNo , x, y, ChangeType)
VALUES (6, sysdate , '333000004', 5, 5, 'C');
INSERT INTO STATE_CHANGE(ChangeId , TimeStamp , PhoneNo , x, y, ChangeType)
VALUES (7, sysdate , '333000001', 3, 3, 'C');
INSERT INTO STATE_CHANGE(ChangeId , TimeStamp , PhoneNo , x, y, ChangeType)
VALUES (8, sysdate , '333000010', 3, 3, 'C');
INSERT INTO STATE_CHANGE(ChangeId , TimeStamp , PhoneNo , x, y, ChangeType)
VALUES (9, sysdate , '333000020', 4, 4, 'O');
INSERT INTO STATE_CHANGE(ChangeId , TimeStamp , PhoneNo , x, y, ChangeType)
VALUES (10, sysdate , '333000020', 4, 4, 'C');
```

Screenshots

PHONENO	X	Y	PHONESTATE
1 333000010	3	3	On

Figura 5 - Tabella TELEPHONE prima delle query

PHO...	X	Y	PHONESTATE
1 333000010	3	3	Active
2 333000020	4	4	On
3 333000001	3	3	Active
4 333000004	5	5	Active

Figura 6 - Tabella TELEPHONE dopo le query

EXID	CELLID	EXCEPTI...
------	--------	------------

Figura 7 - Tabella EXCEPTION_LOG prima delle query

EXID	CELLID	EXCEPTIONTYPE
1	1	1 M

Figura 8 - Tabella EXCEPTION_LOG dopo le query

CHANGED	TIMESTAMP	PHONENO	X	Y	CHANGETYPE
1	1 07-DIC-20	333000010	3	3	O
2	2 07-DIC-20	333000009	15	15	O
3	3 07-DIC-20	333000009	15	15	F

Figura 9 - Tabella STATE_CHANGE prima delle query

CHANGED	TIMESTAMP	PHONENO	X	Y	CHANGETYPE
1	1 07-DIC-20	333000010	3	3	O
2	2 07-DIC-20	333000009	15	15	O
3	3 07-DIC-20	333000009	15	15	F
4	4 07-DIC-20	333000001	3	3	O
5	5 07-DIC-20	333000004	5	5	O
6	6 07-DIC-20	333000004	5	5	C
7	7 07-DIC-20	333000001	3	3	C
8	8 07-DIC-20	333000010	3	3	C
9	9 07-DIC-20	333000020	4	4	O
10	10 07-DIC-20	333000020	4	4	C

Figura 10 - Tabella STATE_CHANGE dopo le query

Punto 3)

- *Evento*: UPDATE OF MaxCalls ON CELL
- *Semantica*: BEFORE / FOR EACH ROW
- *Condizione*: NEW.MaxCalls < OLD.MaxCalls
- *Azione*:
 1. Calcolare il numero di chiamate attive nella cella in cui il valore di MaxCalls è variato
 2. Se le chiamate attualmente attive sono maggiori del nuovo valore di MaxCalls, all'attributo MaxCalls viene assegnato un valore pari al numero di telefono attualmente attivi nella cella in esame

Trigger

```
CREATE OR REPLACE TRIGGER UpdateMaxCalls
BEFORE UPDATE OF MaxCalls ON CELL
FOR EACH ROW
WHEN (NEW.MaxCalls < OLD.MaxCalls)
DECLARE
Calls SMALLINT;
BEGIN
-- Calcolare il numero di chiamate attive nella cella in cui valore di MaxCalls è variato
SELECT COUNT(*) INTO Calls
FROM TELEPHONE
WHERE PhoneState = 'Active' AND x >= :NEW.x0 AND x < :NEW.x1 AND y >= :NEW.y0 AND y < :NEW.y1;
IF (Calls > :NEW.MaxCalls) THEN
-- Se le chiamate attualmente attive sono maggiori del nuovo valore di MaxCalls,
-- all'attributo MaxCalls viene assegnato un valore pari al numero di telefono
-- attualmente attivi nella cella in esame
:NEW.MaxCalls := Calls;
END IF;
END;
```

Query

```
UPDATE CELL SET MaxCalls = MaxCalls -2;
```

Screenshots

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

Figura 11 - Tabella CELL prima della query

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

Figura 12 - Tabella CELL dopo la query

Punto 4)

- *Evento*: UPDATE OF MaxCalls ON CELL
- *Semantica*: AFTER / FOR EACH STATEMENT
- *Condizione*: -
- *Azione*:
 1. Calcolare sulla tabella mutante il numero massimo di chiamate contemporanee effettuabili sulla rete
 2. Se il numero massimo di chiamate contemporanee effettuabili sulla rete è inferiore a 30, annullare l'operazione utilizzando RAISE_APPLICATION_ERROR

Trigger

```
CREATE OR REPLACE TRIGGER ServiceGuarantee
AFTER UPDATE OF MaxCalls ON CELL
DECLARE
SumMaxCalls SMALLINT;
BEGIN
-- Calcolare sulla tabella mutante il numero massimo di chiamate contemporanee effettuabili
-- sulla rete
SELECT SUM(MaxCalls) INTO SumMaxCalls
FROM CELL;
IF (SumMaxCalls < 30) THEN
-- Se il numero massimo di chiamate contemporanee effettuabili sulla rete è inferiore a
-- 30, annullare l'operazione utilizzando RAISE_APPLICATION_ERROR
RAISE_APPLICATION_ERROR(-20000, 'Numero massimo chiamate contemporanee effettuabili sulla
rete < 30');
END IF;
END;
```

Queries

```
UPDATE CELL SET MaxCalls = MaxCalls -1;
UPDATE CELL SET MaxCalls = MaxCalls -10;
```

Screenshots

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

Figura 13 - Tabella CELL prima delle query

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

Figura 14 - Tabella CELL dopo le query

```
UPDATE CELL SET MaxCalls = MaxCalls -10
Report error -
ORA-20000: Numero massimo chiamate contemporanee effettuabili sulla rete < 30
ORA-06512: a "SYSTEM.SERVICEGUARANTEE", line 11
ORA-04088: errore durante esecuzione del trigger 'SYSTEM.SERVICEGUARANTEE'
```

Figura 15 - Eccezione dopo la seconda query