

Sistemi di gestione di basi di dati

Esercitazione 4 – Trigger

Connessione alla base di dati

Connessione via browser Web all'indirizzo <http://130.192.37.58:8080/apex/>

Login

Per autenticarsi collegandosi all'interfaccia Web, inserire i seguenti parametri.

- Nome utente: bdati[scegliere un valore compreso tra 1-100]
- Password: orac[scegliere un valore compreso tra 1-100]

Ad esempio, collegandosi dalla macchina numero 23 del laboratorio, usare come username **bdati23** e come password **orac23**.



Form titled "Login al database" with the instruction "Immettere nome utente e password di database." It contains two input fields: "Nome utente" and "Password", followed by a "Login" button.

Una volta eseguito il login verrà visualizzata una finestra indicante il seguente menu.



Materiale disponibile

Sono disponibili alcuni script contenenti istruzioni SQL per svolgere le seguenti operazioni:

1. creazione della base di dati
2. creazione dei trigger
3. visualizzazione dei trigger
4. modifica della base dati per attivare i trigger.

Gli script sono disponibili:

- sul sito web del corso, nell'archivio ScriptDatabase.zip
<http://dbdmg.polito.it/twiki/bin/view/Public/SistemiDiGestioneDiBasiDati#Laboratori>

Gli script possono essere caricati nell'interfaccia Web selezionando l'icona "SQL" dal menu principale e successivamente la freccia dell'icona "Script SQL" indicata nella seguente figura.



Cliccare sulla freccia a fianco l'icona di "Script SQL" e selezionare la voce carica per caricare lo script. Per eseguire gli script con **Firefox**, impostare la vista "Dettagli" e cliccare sul pulsante "Esegui" relativo allo script da eseguire.

Home > SQL > Script SQL

Script Proprietario BDATI36 Vista Dettagli Visualizza 15 Vai Elimina selezione Carica > Crea >

Modifica

Proprietario

Nome

Autore

Aggiornamento

Ultimo Aggiornamento

Byte

Risultati

Esegui

BDATI36

session_set.sql

BDATI36

3 giorni fa

524

0

BDATI36

drop_index.sql

BDATI36

3 giorni fa

101

0

BDATI36

show_indexes.sql

BDATI36

3 giorni fa

413

0

Comandi utili

Cancellazione di un trigger:

```
drop trigger nome_trigger;  
drop trigger "nome_trigger";
```

Sostituzione (aggiornamento) di un trigger esistente (anziché cancellarlo e ricrearlo):

```
CREATE OR REPLACE TRIGGER nomeTrigger ...
```

Visualizzazione dei trigger generati:

```
select trigger_name, triggering_event, table_name,  
status, description, action_type, trigger_body  
from user_triggers;
```

Disabilitazione di un trigger esistente:

```
ALTER TRIGGER triggerName DISABLE;
```

Visualizzazione degli errori dei trigger:

```
select * from USER_ERRORS;
```

Consigli

Per la creazione dei trigger, prestare attenzione alla sintassi e ai seguenti dettagli:

- assegnare un nome opportuno alle variabili, evitando parole chiave come MIN, MAX, ...
- dichiarare variabili diverse su righe diverse e **non** sulla stessa riga separate da virgola

```
MyVarUno NUMBER;  
MyVarDue NUMBER;  
MyVarTre VARCHAR2(16);
```

- terminare il trigger con il carattere / come nello script `create_triggers.sql` (la creazione del trigger può andare a buon fine anche in assenza del carattere di terminazione);
- terminare le istruzioni con il carattere ; assegnare nuovi valori alle variabili con :=, es.

```
UPDATE tablename  
SET varname=newvalue  
WHERE column=:NEW.attribute;
```

```
IF A<3 OR A=3 THEN  
    MyVar:='Tre';  
ELSE  
    IF A>3 AND A<5 THEN  
        MyVar:='Quattro';  
    ELSE  
        MyVar:='Altro';  
    END IF;  
END IF;
```

- si ricorda che per lanciare un'eccezione/errore è necessario utilizzare la procedura `raise_application_error(error_number, message)`.

Esempio:

```
raise_application_error (-20000, 'Errore durante l'esecuzione dell'operazione xyz');
```

Esercizio

Si consideri la seguente base di dati

```
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)
```

Lo schema descrive una rete di telefonia cellulare. I telefoni all'interno della rete sono descritti nella tabella TELEPHONE. Lo stato di un telefono all'interno della rete può assumere i valori **On** o **Active** (il telefono è Active se sta eseguendo una chiamata). Quando il telefono è spento non è presente nella tabella TELEPHONE. La posizione corrente all'interno della rete di ogni telefono è definita dalle sue coordinate x, y .

La rete di telefonia cellulare è composta da celle. Per semplicità si considerino celle quadrate. Le celle della rete sono descritte nella tabella CELL. La posizione di ogni cella all'interno della rete è definita dalle coordinate x_0, y_0 del suo estremo inferiore sinistro e dalle coordinate x_1, y_1 del suo estremo superiore destro. La cella di appartenenza di un telefono è l'unica cella che soddisfa entrambe le condizioni $x_0 \leq x < x_1$ e $y_0 \leq y < y_1$. Per ogni cella è noto il numero di telefoni correntemente presenti nella cella (attributo CurrentPhone#), e il massimo numero di telefoni che possono effettuare una chiamata (attributo MaxCalls).

Si scrivano i trigger che gestiscono le seguenti variazioni di stato della rete (inserimenti nella tabella STATE_CHANGE).

Punto 1) Accensione/spengimento di un telefono. I tipi di variazione sono 'O' (acceso) e 'F' (spento). Quando il telefono viene acceso, occorre introdurre l'informazione su di esso nella tabella TELEPHONE, mentre quando viene spento occorre eliminarla. Inoltre, occorre localizzare la cella di appartenenza del telefono e modificare il conteggio dei telefoni presenti (attributo CurrentPhone#).

Punto 2) Spostamento di posizione di un telefono. Il tipo di variazione è 'L' mentre x e y rappresentano le coordinate della nuova posizione del telefono. Occorre aggiornare la posizione del telefono. Inoltre, occorre verificare se lo spostamento di posizione del telefono richiede un cambiamento di cella. In questo caso, devono essere modificati i contatori dei telefoni presenti delle due celle interessate.

Punto 3) Inizio di una telefonata. Il tipo di variazione è 'C'. Se la cella in cui è localizzato il telefono non supera il numero massimo di chiamate che può gestire (attributo MaxCalls), occorre modificare lo stato del telefono, che diviene 'Active'. Se invece la cella supera il numero di chiamate che è in grado di gestire, il telefono non può essere attivato. Occorre inserire l'informazione che il numero di chiamate è stato superato nella tabella EXCEPTION_LOG. Il codice ExId è un contatore, univoco all'interno di ogni cella.

Si scrivano i trigger che gestiscono gli aggiornamenti del numero massimo di chiamate gestibili dalle celle (update dell'attributo MaxCalls della tabella CELL).

Punto 4) Variazione del numero di chiamate massime delle celle. Per motivi di gestione della rete di telefonia cellulare può essere necessario diminuire il numero massimo di chiamate gestibili da una cella (diminuzione del valore assunto dall'attributo MaxCalls della tabella CELL). L'operazione di aggiornamento del numero massimo di chiamate gestibili da una cella può portare ad una situazione di inconsistenza in cui il numero massimo di chiamate gestibili da una cella diventa inferiore al numero di telefoni attualmente attivi (PhoneState='Active') localizzati nella cella stessa. Se ciò accade è necessario applicare un'azione compensativa tale da assegnare all'attributo MaxCalls un valore pari al numero di telefoni attualmente attivi presso la cella in esame.

Punto 5) Garanzia del servizio. Il gestore della rete deve garantire un livello minimo di servizio. In particolare è necessario garantire che il numero massimo di chiamate totali contemporanee effettuabili sulla rete (ossia la somma dei valori di MaxCalls considerando la totalità delle celle) non scenda mai al disotto di una soglia minima pari a 30. Eventuali operazioni di aggiornamento della tabella CELL non devono mai portare alla violazione di tale vincolo. Se ciò accade l'operazione di aggiornamento deve essere annullata e deve essere lanciata un'eccezione.

Procedimento:

1. Creare la base dati utilizzando lo script `create_db_telefoni_en.sql`
2. Creare i trigger necessari a soddisfare i requisiti del punto 1
3. Verificare il funzionamento del trigger (telefono acceso) inserendo i seguenti record
 - `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');`
4. Verificare il funzionamento del trigger (telefono spento) inserendo il seguente record
 - `INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType) values (3, sysdate, '333000009', 15, 15, 'F');`
5. Creare i trigger necessari a soddisfare i requisiti del punto 2
6. Verificare il funzionamento del trigger inserendo i seguenti record
 - `INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType) values (4, sysdate, '333000001', 13, 13, 'O');`
 - `INSERT INTO STATE_CHANGE(ChangeId, TimeStamp, PhoneNo, x, y, ChangeType) values (5, sysdate, '333000001', 3, 3, 'L');`
7. Creare i trigger necessari a soddisfare i requisiti del punto 3
8. Verificare il funzionamento del trigger inserendo i seguenti record
 - `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');`
9. Creare i trigger necessari a soddisfare i requisiti del punto 4
10. Verificare il funzionamento del trigger eseguendo il seguente aggiornamento
 - `UPDATE CELL SET MaxCalls = MaxCalls-2;`
11. Creare i trigger necessari a soddisfare i requisiti del punto 5
12. Verificare il funzionamento del trigger eseguendo i seguenti aggiornamenti
 - `UPDATE CELL SET MaxCalls = MaxCalls-1;`
 - `UPDATE CELL SET MaxCalls = MaxCalls-10;`

Durante l'esercitazione dovranno essere eseguiti i seguenti passi:

- *scrivere i triggers;*
- *verificare il risultato di ogni passo controllando i cambiamenti della base dati.*