

Basi di dati attive

Una base di dati è ATTIVA quando consente la definizione e la gestione di regole attive o trigger.

Tali regole vengono attivate in modo automatico al verificarsi di specifici *eventi* sulla base di dati e, verificate alcune *condizioni*, eseguono *azioni* sulla base di dati stessa.

Basi di dati attive

STRUTTURA generale di un TRIGGER:

when <event>

if <condition>

then <action>

<event>: sono eventi che corrispondono a modifiche dei dati (insert, delete, update).

<condition>: è una condizione booleana specificata in SQL.

<action>: è una sequenza di comandi SQL che producono un'azione sulla base di dati.

Basi di dati attive

LIVELLI DI GRANULARITA':

row-level: attivazione per ogni tupla,

statement-level: attivazione una volta sola.

MODALITA' DI VALUTAZIONE e ATTIVAZIONE:

immediata: subito prima/dopo l'evento che ha attivato il trigger,

differita: all'esecuzione del COMMIT.

Basi di dati attive

Sintassi di un trigger in SQL-99

```
CREATE TRIGGER <nome>  
BEFORE | AFTER | INSTEAD OF <evento>  
[REFERENCING <variabili>]  
[FOR EACH {ROW | STATEMENT}]  
WHEN ( <condizione SQL> )  
<sequenza di comandi SQL>
```

Basi di dati attive

<evento>: INSERT ON <tabella> |
DELETE ON <tabella> |
UPDATE [OF <lista_colonne>] ON <tabella>

<variabili>:

OLD AS <nome tuple vecchie>

NEW AS <nome tuple nuove>

OLD_TABLE AS <nome tabella vecchia>

NEW_TABLE AS <nome tabella nuova>

Basi di dati attive

<condizione SQL>: clausola WHERE SQL riferita alle variabili definite o di default.

<sequenza di comandi SQL>: se si tratta di un BEFORE trigger allora la sequenza di comandi SQL è limitata alla modifica delle variabili NEW (NEW_TABLE), altrimenti è una qualsiasi sequenza di comandi SQL, racchiusa tra BEGIN ATOMIC e END, che può riferirsi anche ad altre tabelle della base di dati (è disponibile l'istruzione SET per modificare la variabile NEW).

Basi di dati attive

Si noti che le politiche di reazione alla violazione dei vincoli di integrità referenziale specificabili in SQL:

```
<vincolo>.... on {delete | update}
    {cascade | set null | set default | no
action}
```

possono essere visti come trigger predefiniti che reagiscono alla cancellazione o alla variazione di una chiave esportata.

Basi di dati attive

Meccanismo di attivazione dei trigger (escluso il trigger **INSTEAD OF**).

Data una primitiva P INSERT/DELETE/UPDATE:

1. Si eseguono i trigger before statement attivati da P
2. Per ogni tupla della tabella coinvolta in P:
 - a) Si eseguono i trigger before row attivati da P;
 - b) Si esegue P e i test per la verifica dei vincoli di integrità (potrebbero attivare altri trigger after)
 - c) Si eseguono i trigger after row

Basi di dati attive

3. Si eseguono i test di verifica dei vincoli di integrità che devono valere sull'intera tabella.
4. Si eseguono i trigger after statement

L'attivazione di nuovi trigger viene gestita in modo simile all'attivazione di procedure ricorsive.

I trigger before possono solo modificare le variabili NEW e non possono attivare altri trigger.

Basi di dati attive

Esempio di trigger in SQL-99:

Studente(matricola, nome, anno)

Esame(matricola, insegnamento, data, voto)

```
CREATE TRIGGER T1
```

```
  AFTER UPDATE ON Studente
```

```
    REFERENCING NEW AS n
```

```
    FOR EACH ROW
```

```
  WHEN (n.anno > 2 AND (SELECT count(*) FROM
```

```
    Esame WHERE matricola = n.matricola ) <= 7)
```

```
  ROLLBACK;
```

Proprietà delle regole attive

Osservazione: la progettazione di una regola attiva è semplice, più complesso invece è comprendere il comportamento collettivo di un insieme di regole attive.

Proprietà di un insieme di regole attive:

- Terminazione
- Confluenza
- Determinismo delle osservazioni

Proprietà delle regole attive

Terminazione

Un insieme di regole garantisce la terminazione quando, per ogni transazione che attiva le regole, l'esecuzione TERMINA producendo uno stato finale.

Proprietà delle regole attive

Confluenza

Un insieme di regole garantisce la confluenza quando, per ogni transazione che scatena l'esecuzione delle regole, l'esecuzione termina producendo un UNICO stato finale che non dipende dall'ordine di esecuzione delle regole non esplicitamente priorizzate.

Proprietà delle regole attive

Determinismo delle osservazioni

Un insieme di regole garantisce il determinismo delle osservazioni quando, per ogni transazione che scatena l'esecuzione delle regole, l'esecuzione è confluyente e tutte le azioni svolte dalle regole sono identiche e prodotte nello stesso ordine.

Test di terminazione

- Costruire il grafo di attivazione per l'insieme di regole R:
 - NODI: coincidono con le regole dell'insieme R
 - ARCHI: esiste un arco da r_i verso r_j quando un'azione di r_i contiene una primitiva che coincide con uno degli eventi che attivano r_j .
- L'esecuzione delle regole di R PUO' non terminare SOLO in presenza di cicli nel grafo di attivazione di R.

Esempio

CREATE TRIGGER ControllaSalari

AFTER UPDATE OF salario ON Impiegato

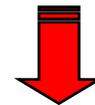
FOR EACH STATEMENT

WHEN (SELECT avg(salario) FROM Impiegato) >
100

UPDATE Impiegato SET salario = 0.9 * salario;



GRAFO CICLICO



Non terminazione possibile